A Web server for accessing a database on solar radiation parameters

Joel Angles, Lionel Ménard, Olivier Bauer, Lucien Wald

Published in: J. STROBL and C. BEST (Eds.), 1998: Proceedings of the Earth Observation & Geo-Spatial Web and Internet Workshop '98 = Salzburger Geographische Materialien, Volume 27. Instituts für Geographie der Universität Salzburg. ISBN: 3-85283-014-1

A Web Server for Accessing a Database on Solar Radiation Parameters

J. Angles, L. Ménard, O. Bauer, L. Wald

Ecole des Mines de PARIS, Centre d'Energétique, Groupe Télédétection & Modélisation
BP 207 06904 Sophia Antipolis Cedex
menard@cenerg.cma.fr
http://www-helioserve.cma.fr/

Contents:

Keyword:

Solar energy, atmosphere, mSQL, PHP/FI, Apache.

1. Abstract

Ecole des Mines de Paris has gathered since many years a lot of Earth Observation data of different kinds and is willing to provide those data to a broad audience.

In collaboration with European partners, Ecole des Mines de Paris has assembled a dataset, which is unique in the world. It deals with one parameter of the solar radiation: the Linke turbidity factor, which characterizes the opacity of the atmosphere.

Recently, the advent of the World Wide Web (WWW) [1], has laid the foundation of a simple, friendly and uniform way of sharing data on internet. Nevertheless, the complexity and the size of some dataset and the heterogeneity of files formats were not suited to be put on-line on the web without preparation.

The need of being able to make sophisticated queries to extract relevant information has therefore lead us to build a database (SQL) / Web server.

The HELIOSERVE server has been opened in late November 1997. Several customers have already praised its usefulness. This paper explains the design of the web server call HELIOSERVE. It describes the scientific context and explains the different steps to transform a dataset into a database that can be queried via the WWW.

The authors have also examined a more general problem : the construction of a web server by small university-like entities which want to promote their database in Earth Observation related domains. Some conclusions were reached and are presented. Guidelines on how to help these small size providers are discussed.

Back to the contents

2. Scientific context

Solar radiation is of large importance in many fields, ranging from climate to agriculture or building architecture. Several programs are on going which aim at mapping solar radiation parameters for several parts of the world, including Europe. These programs require tremendous amounts of work. They usually involve the collection of measurements made at ground level and their processing, including quality control. Images of the satellite Meteosat are used in combination with the ground measurements to do the mapping.

The Linke turbidity factor (T_L) [2] is a very convenient approximation to model the atmospheric absorption and scattering of the solar radiation under clear skies. T_L describes the optical thickness of the atmosphere due to both the absorption by the water vapor and the absorption and scattering by the aerosol particles relative to a dry and clean atmosphere. It summarizes the turbidity of the atmosphere, and hence the attenuation of the direct beam solar radiation and the importance of the diffuse fraction. The larger T_L, the larger the attenuation of the radiation by the clear sky atmosphere. The smallest value of T_L is 1.0, obtained for a fully dry and clean atmosphere. T_L generally lies between 2.0 (dry and clean atmosphere) and 6.0 (humid and polluted atmosphere).

The Linke turbidity factor is an important parameter in solar radiation studies. Ecole des Mines de Paris has assessed monthly values of T_L for a total of 595 sites over Europe and countries surrounding the Mediterranean sea. This constitutes by far the largest database of T_L in the world.

Figure 1: Maps of the site where T_L values are provided

The customers of the HELIOSERVE site are presently researchers deeply involved in studies of solar radiation parameters. They precisely know what is T_L, and the database outputs act as inputs into their own models for atmospheric optics. A larger class of customers is targeted. It is made up of professionals, in research or industry, who want an assessment of the maxima of the solar radiation available at ground level, for e.g., the sizing of a solar system for domestic water heating. Such customers do not care of values of T_L, though it is an important parameter for the assessment of the solar radiation for clear skies. For them, numerical simulation has been developed. The final outputs are then values of available solar energy.

Back to the contents

3. HELIOSERVE design

The HELIOSERVE server was built up by taking into account two aspects:

1. A database query aspect. It is the ability of the server to supply the customer with the T_L values, which are extracted from the database.
2. A numerical simulation aspect. It provide to the customer estimates of the clear sky radiation by running a model. Some of the inputs are the T_L values, which are supplied either by the database, or by the customer himself, or set to default

values.

HELIOSERVE relies on a set of softwares that are called by an http server in response to a user requests. The T_L values are stored into a relational database. The system interface allows a user to send a query to the web server through HTML (HyperText Markup Language) [3] documents based on a form [4]. This HTLM page calls a template that includes SQL statements as well as conditional instruction and HTML tag for the output.

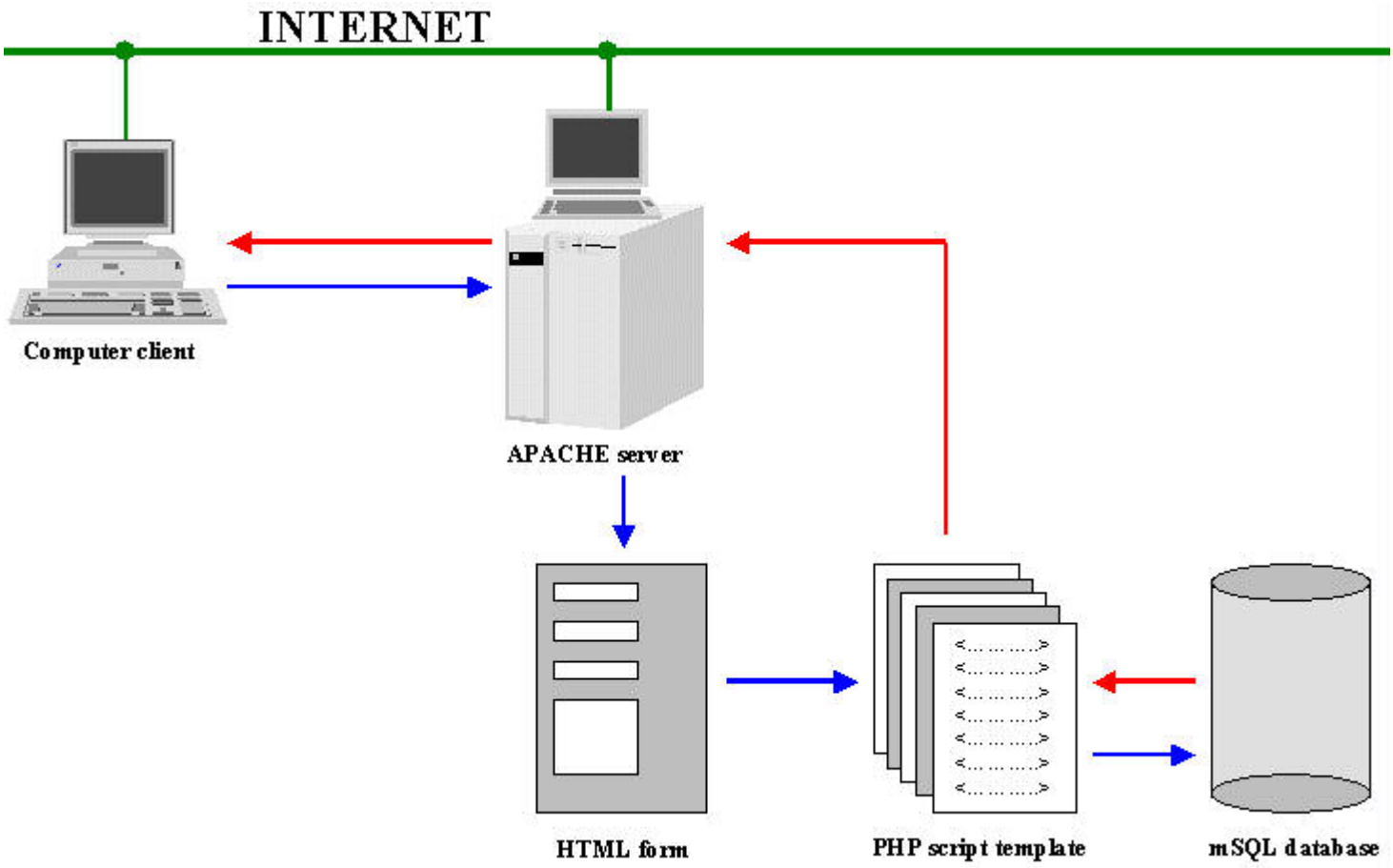

Figure 2: Overview of the HELIOSERVE server

The choice of UNIX as the O.S. (Operating System) for the development of HELIOSERVE was obvious for two reasons. Firstly many people in our institution are fluent with this O.S. Secondly considering that this project was self-funded we have tried as much as possible to use free software and the UNIX community offers the broadest choice.

The HELIOSERVE server uses an APACHE [5] web server. The APACHE server is powerful and flexible. It is highly configurable and allows extension to third party modules. It is the most used server with more than 45% of the overall web server on internet [6]. It is part of a huge consortium deeply committed into its development and maintenance. Moreover it is free.

As a database manager, mini SQL (mSQL) [7] was chosen. This relational database package is free for educational bodies, such as Ecole des Mines de Paris.

The mSQL database only allows a subset of ANSII SQL query language but we did not yet face any limitation for our project. As mentioned by their authors the philosophy of mSQL has been to provide a database management system capable of rapidly handling simple task. mSQL also includes various tools like a C API allowing any C program to communicate with the

database, a www interface, called W3-mSQL, and a scripting language called " Lite ".

The missing link of the system is the scripting language that will handle the HTML-based form query of the user to the mSQL database.

Despite the above mentioned "Lite" scripting language included with mSQL, we have preferred another tool for our script development. This tool is PHP/FI [8].

We find PHP/FI to have more features for database queries, to be more documented, and to offer a larger set of example of successful web / database server application.

PPH/FI is a server-side html-embedded scripting language. That means that a simple script can be written directly into html files. Moreover it is not browser-dependant. It include support for various databases like (mSQL, Postgres, MySQL, Sybase, Oracle...).

One of the important features of PHP/FI is that it can be bundled as an Apache module, allowing a fast and powerful alternative to CGI programming. This is what has been done for HELIOSERVE.

Summarizing some of the advantages are :

Performance : PHP/FI does not act as a standard CGI program since the script code is executed directly by the Apache web server process.

Configurability : The parser is configured on start-up in the same configuration file as the httpd server process.

Security : When running PHP/FI as a module the httpd based access restriction is defined in the httpd server configuration file.

Custom function : A framework for writing your own module is offered.

Back to the contents

---

4. Database design

The database query aspect is the ability of the HELIOSERVE server to supply the customer with the T_L values. These values are extracted from the database.

The database is constituted by a list of geographical sites with assessment of the Linke turbidity factor. These stations are spread throughout Europe. For each station, monthly values of the Linke turbidity factor are available, that is twelve values per station. The database comes from two Microsoft Excel format files:

A file taking stock characteristics about the station subdivided into nine fields (Figure 3):

1. specific number or WMO (World Meteorological Organization) number for this station
2. station name (stationname)
3. latitude in hundredths of decimal degree (lat)
4. longitude in hundredths of decimal degree (lon)
5. elevation in meter (elevation)
6. referenced country code (countcode)
7. country (countryname)
8. line in the reference map in raster format (line)

9. row in the reference map in raster format (row).

WMO	Stationame	Latitude	Longitude	Elevation	Ref. Country	Country	Line
01316	Bergen	6040	532	41	NORWA	NORWAY	176
02045	Kiruna Geofysika	6783	2043	408	SWEDE	SWEDEN	87
02185	Lulea	6555	2213	17	SWEDE	SWEDEN	114

Figure 3: Stock characteristics of the stations.

A file which contains the specific number of every station and twelve Linke turbidity factors per station (Figure 4).

WMO	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
01316	2.78	3.43	3.23	3.42	3.33	2.7	3.28	3.23	2.68	2.97	3.05	2.85
02045	3.53	4.09	4.31	3.42	3.33	2.87	2.91	3.06	2.83	3.87	4.07	3.33
02185	2.78	3.96	3.7	3.42	3.88	3.55	4	3.57	3.43	3.61	3.16	2.56

Figure 4: Unique WMO number for each station and twelve T_L per station.

4.1 mSQL database language

mSQL has a set of programs that allow to build, customize, administrate and query a database. It has also an import and export function for various file format.

Here is the syntax used to create the *caract2* table in the Linke database.

```
create table caract2 (wmo int,stationname char(40), lat int,lon int,elevation
int,countcode char(5), countname char(40), line int, row int )
```

The `relshow` command of mSQL allows to display the structure of the Linke database.

Database = linke

Table = caract2

Field	Type	Length	Not Null	Unique Index
wmo	int	4	N	N/A
stationname	char	40	N	N/A
lat	int	4	N	N/A
lon	int	4	N	N/A
elevation	int	4	N	N/A
countcode	char	5	N	N/A
countryname	char	40	N	N/A
line	int	4	N	N/A
row	int	4	N	N/A

After building the structure and instead of entering manually the data into the table we used the `import` function of mSQL. We first changed the Microsoft Excel file format into an ASCII file where all the field were comma separated.

When the database is built up queries can be issued. mSQL allows to execute command line query. This is a good way to get familiar with the SQL syntax before trying to do more complex scripting. Here is the syntax and some queries as an example.

```
SELECT [table1.field],[table2.field],...

FROM [table1],[table2],...

WHERE [table1.field] OPERATOR VALUE

AND/OR [table2.field] OPERATOR VALUE

ORDER BY [table(1/2).field]
```

The operators used are <, >, =, <=, >=, <>, CLIKE. Most of the time, VALUE is a literal value.

The first queries for testing are simple. It is of interest to query simultaneously the caract2 and coeff tables. The example below show a "select" query with some parameters, and the resulting table.

```
select coeff.mar,caract2.countryname from caract2,coeff

where caract2.lat>=5823 and caract2.lon<1200 and caract2.wmo=coeff.wmo \g

+--------------+-----------------------------------------+

| mar | countname |

+--------------+-----------------------------------------+

| 3.08 | UNITED KINGDOM |

| 3.08 | UNITED KINGDOM |

| 2.93 | UNITED KINGDOM |

| 2.46 | UNITED KINGDOM |
```

```
| 3.23 | NORWAY |

| 2.31 | ICELAND |

| 2.31 | ICELAND |

| 2.31 | ICELAND |

| 2.16 | ICELAND |

+--------------+-----------------------------------------+
```

Once familiar with the SQL queries into mSQL, it is possible to build dynamic HTML pages and to include SQL query and all kind of conditional statements. This can be done by using the scripting tool PHP/FI.

4.2 PHP/FI script language

Most of the main characteristics of PHP/FI have been given above. This section discusses how it can be practically used to build scripts embedded into HTML pages.

The example below shows the code embedded into real simple HTML tags. This page when launched via a browser, displays the WMO and the station name for the measurement station with an elevation between 145m. and 150m.

```
        <html>

        <body>

        <center>

                <?

        $y=100; $o=145; $p=150;

        $result = msql(" linke ", "select * from caract2 where caract2.elevation>$o
        and

        caract2.elevation<$p order by caract2.wmo ");

        $num = msql_numrows($result);

    /* Error test */

        if ($num==0);

        echo "  Sorry but no record found!<p> ";

        else;

        echo " $num records found!<p> ";

        endif;

    /* End of error test */

        $i=0;

        while($i<$num);
```

```
echo " <b> ";

echo " stationname= ";

echo msql_result($result,$i, "stationname ");

echo " <br> ";

echo " wmo= ";

echo msql_result($result,$i, "wmo ");

echo " </b> ";

$i++;

endwhile;

       >

</center>

</body>

</html>
```

The next step will be to build some HTML page that will include form in order to let the user build its own query. For the HELIOSERVE project and regarding the database complexity we build six different HTML pages including forms. This is where it is getting really interesting with the scripting process as users can issue endless queries to your database when you only need to write a small set of templates. That means that there is no static HTML pages stored into the web server. All the pages are dynamically built upon user request.

The templates building process is a good way to proceed because a file will be generated that will contains both the script code and the formatted HTML to be output. This is also very useful because all the code developed in this template is hidden, and cannot be seen by the user.

The tricky thing about PHP/FI as bundled as an APACHE module is that all the file to be built MUST have a '*.phtml*' extension. As soon as the APACHE server see a '*.phtml*' extension, it directly process it. It is much faster than the classic CGI scheme.

To summarize, the PHP/FI scripting process, requires:

1. To create a form based '*.phtml*' page to supply the queries
2. To create a '*.phtml*' template that will include the script code and the HTML tags to output.

For HELIOSERVE we have build six different form-based '*.phtml*' pages. We have also split the result into two pages because it was most convenient for reading. The intermediate page is dynamically built by PHP/FI and does not contain the result itself but a hyperlink to it.

Below are three figures illustrating the steps of querying the HELIOSERVE server.

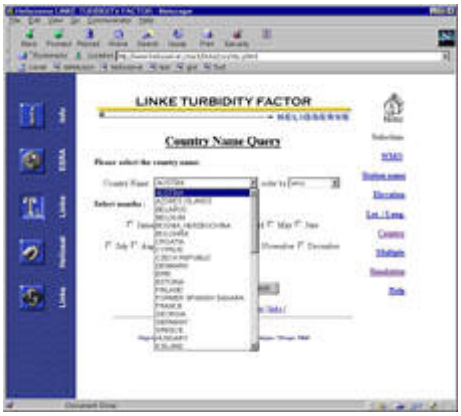

Fig. 5:Query page

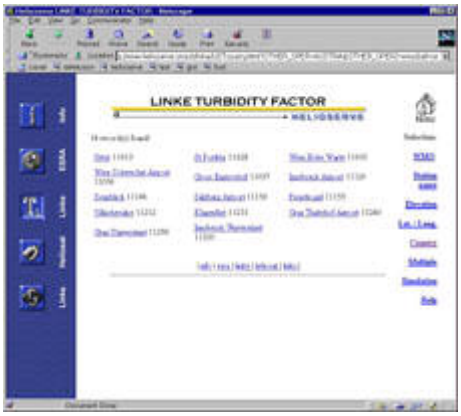

Fig. 6: Intermediate result

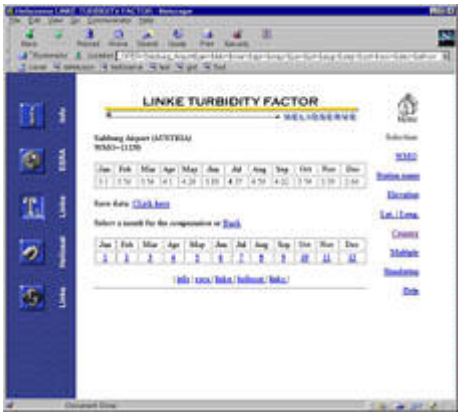

Fig. 7: Final result

4.3 The numerical simulation

The numerical simulation aspect provides to the customer, estimates of the clear sky radiation by running a model.

This has been made using PHP/FI and a third party simulation software. The simulation software was written in C language and has been tailored to suit the web application. Modifications have only been made on the output section of the source program. The simulation can be executed at two different places into HELIOSERVE involving different schemes.

1. It fetches some parameters from the result of a query made by the user, and asks for two more parameters (the day and the year) for computation (figure 8)
2. Or the user can manually feed the empty fields with his own parameters (figure 9)

The result is displayed into the user's browser as shown in figure10.

Fig. 8: Database simulation

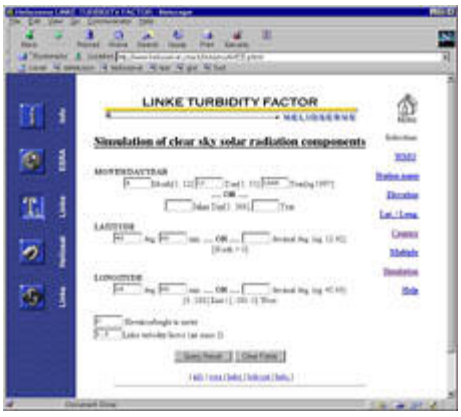

Fig. 9: Simulation page

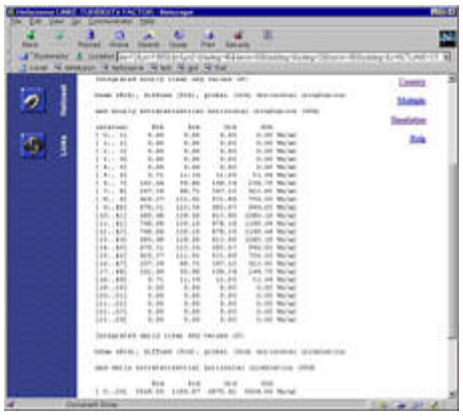

Fig. 10: Simulation result

Back to the contents

5. Exploitation results

HELIOSERVE has been opened in late November 1997. Around 2000 requests of foreign computers from more than 20 different countries have been recorded into the server logs file at the end of January 1998. This may appear very small but the user community involved in the use of solar radiation and specially Linke turbidity factor is not widespread and HELIOSERVE has not been extensively published. It is still in a test phase for its scientific contents as well as for the user interface. Many users have already praised its existence because it is currently the easiest and cheapest way to obtain the Linke turbidity factor as well as simulations of the available solar energy under clear sky.

The whole system has also proved its reliability. It is simple to administrate and new features can be added like new templates without any repercussion on the overall system.

Back to the contents

6. Conclusion

The availability of free software has enabled the authors to build up a web server giving access to a scientific database unique in the world. This server HELIOSERVE also provides results of numerical simulations. It is likely to evolve and to offer more information on solar radiation and also related software. The database is very simple and most of the efforts have been put on the management of the queries through the web as well as on the user interface. Though our team may be considered as trained in all the aspects related to the HELIOSERVE server, the amount of efforts was more important than expected. The virtue of an example is very important and it is likely that further improvements of HELIOSERVE or the development of new servers will require less efforts by our team. However it points out to a presently weak aspect in the promotion of the Earth Observation through internet. It is obvious that the usefulness of internet in Earth Observation depends upon the quality of the information made available to the community. Besides several large providers (e.g., NASA, SPOT-Images, ...) which have already make efforts, large amounts of very valuable information are within small university-like entities. These entities are willing to promote their databases or algorithms but because of their small size, cannot presently afford even simple developments of web server such as HELIOSERVE. In this context, the Earth Observation community must praise the initiatives of the Center for Earth Observation (CEO) of the European Community, which offers several tools (toolkits, frames, standards) to help these small providers to build web servers. These initiatives are quite recent and have not been widely publicised. Belonging to an educational body, the authors stress the importance of training and urge, e.g. the CEO, to set up courses to teach the small providers how to build a server to promote their data. This would increase the use of Earth Observation-related information and would benefit both the scientific and non-scientific communities.

Back to the contents

7. Acknowledgments

The database has been constructed in co-operation with MM. Gerhard CZEPLAK (Deutscher WetterDienst, Hamburg, Germany) and John PAGE (Sheffield, United Kingdom).

Back to the contents

8. References

[1] The World Wide Web Consortium: http://www.w3.org/

[2] The Linke turbidity factor (T_L): http://www-helioserve.cma.fr/linke/publication.html

[3] HyperText Markup Language: http://www.w3.org/MarkUp/

[4] Mosaic for X version 2.0 Fill-Out Form Support:

http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/Docs/fill-out-forms/overview.html

[5] The Apache HTTP Server Project: http://www.apache.org/

[6] The Netcraft Web Server Survey: http://www.netcraft.com/survey/

[7] MiniSQL (mSQL): http://www.Hughes.com.au/

[8] PHP/FI home page: http://php.iquest.net/

Back to the contents
