PERSPECTIVE ON REMOTE ACCESS TO SPACE ENVIRONMENT MODELS AND DATABASES

M. Kruglanski[(1)], E. Speelman[(1)], B. Quaghebeur[(1)], D. Heynderickx [(1)]

[(1)]*Belgian Institute for Space Aeronomy, Ringlaan 3, B-1180 Brussels, Belgium, m.kruglanski@oma.be.*

ABSTRACT

Quantifying the space environment requires retrieval of spacecraft or ground based data and access to empirical or physical models. Many data providers are operational world wide, but they employ various methods and forms of data delivery and it is often a daunting task to locate, retrieve, extract and re-format the relevant data. Services providing remote access to space environment models are not so numerous (the SPENVIS [1] system is one example of such a service). In the framework of a prototyping activity, part of an ESA space weather programme study [2], an integrated system has been developed to access in a uniform way both models and databases [3]. Results of this prototype activity are presented as well as some suggestions to standardize remote access to space environment data and models.

1. INTRODUCTION

When heavy computational processes or large amounts of data are addressed, it can be valuable to share computer resources. In a distributed environment where users are located at different places, even different countries, such sharing requires the implementation of remote access to databases and program codes. In this paper, we focus on remote access in the framework of space environment applications. In particular, we consider the case of occasional or not-specialized users, and the usage of physical models in continuous development.

At the present time, functional implementation of remote access to space environment applications are not so numerous and have different implementation strategies. In Section 2, we present a short review of some existing implementations. In Section 3, we propose the design of a general interface for remote access and discuss its implementation. In Section 4, we present our conclusions, focusing on some open issues.

2. REVIEW OF SOME EXAMPLES

In this section, we present four examples of remote space environment models. The first two examples are functional cases freely accessible through the World Wide Web (WWW). The last two examples are working demos developed in our institute.

2.1 <u>SPENVIS models</u>

The SPENVIS [1] system provides access to engineering-type models of the space environment and its effects on space systems through a multi-page WWW interface. The system includes models addressing space radiation analysis (due to trapped particles, solar protons and cosmic rays), spatial and temporal patterns of the magnetosphere, atmosphere and ionosphere, spacecraft charging, etc. Most models can be accessed using several kinds of time-position inputs, such as spacecraft orbit, world map and vertical profile. The inputs also include model parameter control, e.g. *Dst*, solar wind density and velocity for external magnetic field of Pfitzer et al., 1988 [4]. The model results are provided in both graphical and text formats.

One of the features of this system is the possibility of model grouping, whereby a suite of models stores and reads intermediate results, without the need for user interaction. For instance, when evaluating solar cell degradation with the EQFRUX model [5], SPENVIS automatically retrieves particle fluences from the results of a trapped radiation belt model and/or a model of solar proton events.

The SPENVIS system is only suited for human users, as opposed to automated access, since the interface requires passing through different pages to provide the input parameters, to execute the models and to download their results.

2.2 <u>Lund forecasts</u>

The Lund division of the Swedish Institute of Space Physics has developed a prototype of a space weather forecasting service based on neural network computing [6]. The service already provides few-hour forecasts for both the planetary magnetic index (*Kp*) and the disturbance storm-time index (*Dst*). The forecast algorithm uses, as input, ACE solar wind data automatically downloaded from NOAA's Space Environment Center in Boulder. The prediction results are obtained in the form of graphics through a web interface. No input parameters are required.

2.3 ESWS prototyping activity

In the framework of the ESA space weather programme study lead by the Rutherford Appleton Laboratory [2], we have developed a prototype web-based interface that includes yellow pages linked to a data retrieval tool and a space environment model access tool. The yellow pages contain detailed references to existing space or ground missions and instruments that provide public space environment data. Subsequent to a user query, the retrieval tools are able to automatically locate and download files from the Internet, to extract the requested data, to combine all the requested data into a single file, and to produce a graphical preview. As an example, the data stored in the yellow pages for one *channel* are presented in Table 1. In the framework of the prototype, a channel is defined as a time-series value both associated to a sensor and to a "column" in a file. The example refers to 5-minute averaged GOES-8 proton integral flux that can be retrieved from the NOAA Space Environment Center. Inside the system, the data are shared over several linked catalogues. For the sake of clarity, some fields have been merged and some values abridged. When a user requests this channel, the system expands the generic URL according to the requested time range and downloads the corresponding files. When the channel is associated to a space mission, ephemeris data are downloaded as well. Channels from different instruments and missions can be requested in a single query.

Access to external models is implemented in the prototype by associating some channels to a model instead of to an instrument and mission. When selecting such a channel, the user has to specify a spacecraft, whose ephemeris will be used as model input. The external models are supposed to be accessible through a WWW-server script to which the

Table 1: Example of the data stored in the yellow pages for a channel referring to 5-minute averaged GOES-8 proton integral flux archived in an ASCII file on the FTP site at NOAA

Catalogue	Field	Value
Provider	Description	Space Environment Center anonymous FTP system
	Manager	E. Hildner
	Contact	Mrs. V. Raben (vraben@sec.noaa.gov)
	Institute	SEC: NOAA Space Environment Center
	Web site	http://www.sec.noaa.gov/
	Address	Code E/GC2, 325 Broadway, Boulder, CO 80303-3328, USA
Source file	Description	5-minute GOES-8 Solar Particle and Electron Flux Channels (ascii file)
	Generic URL	ftp://ftp.sel.noaa.gov/pub/lists/particle/***yyyymmdd***_G8part_5m.txt
	Accessibility	from 46 days ago to yesterday
	Time resolution	300 seconds (or longer)
	Nbr. of columns	15
	Extraction prgm.	`extr_ascii_file;skip=2,comm="#",pass=2,sepa=" ",t_co=…`
Channel	Column	11
	Dependences	Proton energy: 50 – 700 [MeV]
	Component	None (scalar data)
Parameter	Description	Unidirectional integral proton flux
	Unit	cm^{-2} sr^{-1} s^{-1}
Sensor	Description	GOES-8 Energetic Particle Monitor: The EP8 sensor consists of detectors …
	Time resolution	10.2 seconds (or longer)
	P.I.	Dr. H. H. Sauer (hsauer@sel.noaa.gov)
	Institute	NOAA Environmental Research Laboratories
	More…	http://rsd.gsfc.nasa.gov/goes/text/databook/section05.pdf NSSDC Master Catalogue (entry 1994-022A-5)
Mission	Description	NASA/NOAA Geostationary Operational Environmental Satellite – 8
	Supervisor	Program Scientist: Dr. R. D. Zwickl (e-mail unknown)
	Institute	SEL/NOAA Environmental Research Laboratories
	Reference	GOES I-M DataBook, Space Systems Loral, DRL 101-08 Revision 1, GSFC Ref…
	Last TLE	`1 23051U 94022A   02037.30282740 -.00000256  00000-0…`
	Ephemerides	Pointer to the *source file* entry: GOES-8 Daily position from SSCWeb Locator (html file) http://sscweb.gsfc.nasa.gov/…?SPCR=goes8&START_TIME=***yyyy***/***mm***/***dd***&…

ephemeris file is uploaded. An example of such a web-script can be found at the URL http://eve.oma.be/ESWS/externals/. The script allows the online evaluation of the NASA models AE-8 and AP-8 MIN and MAX [7, 8], the AFRL CRRESPRO active and quiet models [9], and the MSIS-90 model [10]. It requires, as input, a four-column ASCII file with a set of times and positions. Time is expressed in days since January 1, 1950 and positions are specified by x, y, z geographic Cartesian coordinates in km. The script result is a column-oriented text table.

2.4 SPENVIS/REAT development

In the framework of SPENVIS, it is planned to implement remote access to models developed in the Radiation Effects Analysis Tools (REAT) project [11], i.e. a suite of Geant 4 Monte Carlo transport codes simulating space radiation effects in electronics and other hardware. Since the REAT models are computer intensive, a remote and asynchronous execution is preferred. Two successive HTTP connections are used to implement such an execution. The first connection, initiated by the SPENVIS system, activates the model execution. The model hosting system generates the second connection to send the execution results back to SPENVIS. No connection is kept alive during the model execution.

On the hosting system, the model interface is realized by a small WWW-server script written with PHP [12]. The script assumes that the model codes are batch-oriented. Its configuration includes for each model:

- the input file names,
- the output file names,
- a file name generated by the model when an error occurs that aborts the programme execution,
- a set of environment variables of the operating system that can be modified by a user before the model run, and,
- the model execution command line.

When activated, the script requires, as input, in addition to the model input files and the possible system environment variables:

- a unique request ID to be sent back with the results,
- an email to be used in case of unrecoverable errors,
- the URL to call on model completion in order to send the model results, and,
- the model name.

After verification of the script input, the HTTP connection is ended and the model is executed. On model completion, an HTTP connection to the specified URL is established to transfer the model output files, together with the request ID, the email and a model execution status string. Both HTTP connections use the POST method with file upload [13].

The system has been successfully tested with a module of the CREME model suite [14] that evaluates the LET spectrum from the radiation environment experienced by a spacecraft along its orbit. The implementation didn't require a modification of the model code. The system showed to be well tailored for computer intensive models. As a drawback, human users need to go through a third-party service such as SPENVIS, since the script presumes a HTTP daemon on the requester side to handle the model results.

3. SCENARIO FOR A GENERAL INTERFACE

The introduction of an interface layer clearly separates the request side from the model execution in order to minimize the need of dedicated development. It allows users to access models in a similar way and it prevents developers from having to manage different execution procedures. Such an interface is essential in the field of the space environment where, a significant fraction of the requests will emerge from occasional and/or not-specialized users [15], and models are continuously under development.

When accessing a new model through a standardized interface, the user is guaranteed that he will benefit from his past experience gained from other space environment model executions. His learning curve is mainly driven by the interface instead of by the model implementation. Moreover the user endeavour is greatly reduced if the interface is based on familiar desktop computer tools. The adoption of a common interface also benefits the model developers by reducing their time spent on user input/output handling.

We propose an asynchronous request/response scenario that uses well-established WWW protocols and which includes an option of delayed data exchanges. Our scenario, illustrated in Figure 1, is divided in four successive parts: (i) request submission, (ii) input pre-processing, (iii) model execution, (iv) result transmission. This scenario could possibly be implemented using XML-RPC format [16] or SOAP messages [17, 18]. However, we choose not to adhere to these standards since they require the installation of dedicated software at the user's side.

3.1 Request submission

The underlying protocol to submit the request could be either HTTP or SMTP, i.e. web form or email. The request shall include a unique identifier to correlate the response with the originating request, the email address identifying the sender, the name of the target model, and the URL for the result routing. The routing URL shall correspond to either a WWW-server script or an email address. When the selected model requires input

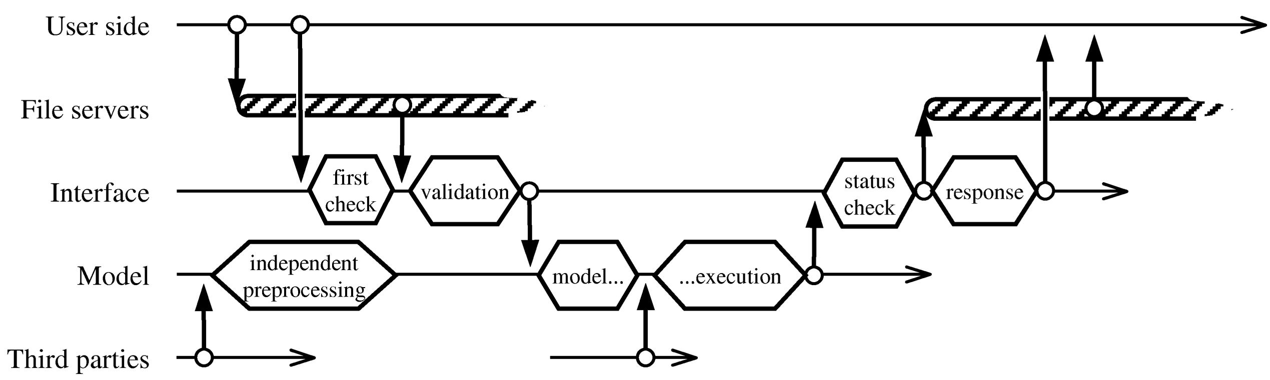

Figure 1: Flow diagram of the proposed request/response scenario for remote access of space environment models.

files, the request shall include a list of filename-URL pairs. The URLs could point either to a file attached to the request by using the content identifier scheme (`cid:…`) [19] and a MIME multipart [20] structure or to an external resource (e.g. a file on a FTP server). Leaving part of the input files out of the request submission leads to a drastic reduction of the request size that can be crucial in the case of email. Since both HTTP and SMTP protocols support an error mechanism, a simple validation can be supplied by the interface in order to reject invalid requests.

3.2 Input pre-processing

Those input files that are not attached to the request, shall be downloaded by the interface from the specified remote servers. The interface shall support HTTP GET, HTTP POST and FTP methods as well as basic user authentication. Note that generic syntax of an URL [21] permits to specify a user/password pair and to pass additional parameters, e.g. "`method=post`" or "`mode=ascii`".

When a file retrieval has failed or a mandatory input file is missing, the interface shall prevent the model execution.

3.3 Model execution

The interface shall initiate the model execution in a way similar to a batch submission. We suggest the creation of a temporary directory where user's inputs and results are located and the model is executed. It is possible that the model needs additional input, which are not delivered by the user. Examples of such input are *Dst*, solar wind density and velocity for Pfitzer et al. model [4], or, the ACE solar wind data for the Lund forecast of *Dst* [6]. Such input should be retrieved through an automatic process independent of the interface.

A simple algorithm shall allow the interface to determine whether the model run was successful or not, e.g. presence or absence of a specific file.

3.4 Result transmission

Whether the model has run successfully or not, a response shall be generated for each valid request accepted by the interface. The response is directed to the routing URL specified during the request submission. In case of trouble during the response connection, a warning email should be sent to the initial sender. In addition of the model results, the response shall include the execution status and the four key elements of the initial request: the unique identifier, the sender email, the model name and the routing URL. The model results should be sent in a similar way as the input files: by a set of filename-URL pairs. For large result files, we strongly suggest the use of an FTP/HTTP server to temporarily store the result files in order to limit the response size. Such a practice could also be valuable when a model output can be used as input to other models.

For the execution status, we suggest to adopt a string starting by a 3-digit code, as it is the convention with both HTTP and SMTP protocols.

4. CONCLUSION

Referring to our past and current involvements in space environment modelling [1, 3, 22], we advocate the definition of a standard interface to remotely access space environment models and databases. Therefore we propose a request/response scenario based on the review of some functional or pilot examples. The main advantages of this scenario are its use of well-established WWW protocols, the absence of dedicated user applications, optional delayed data exchanges, and no connection during the model execution. However, such a scenario is useless without it being adopted by the model developer and user community.

Some open issues are not addressed by the proposed scenario. One major issue concerns the format of the input and result files, and the description of their contents. While some standardized file formats are already applied in the space community, a lot of progress is required in the context of data description. This issue is important when considering model grouping where a remote model could accept, as input, the result produced by an other distant model without the need for file translation.

5. REFERENCES

1. Heynderickx D., Quaghebeur B. and Speelman E., *The Space Environment Information System – Spenvis*, Space Environments and Effects Final Presentation Days, ESTEC, Noordwijk, 14-15 December 1999.
2. Hapgood M., *Study overview*, ESA Space Weather Programme Study Final Presentation Days, ESTEC, Noordwijk, 6-7 December 2001.
3. Kruglanski M. and Heyndrickx D., *Space Weather Prototyping Activity*, ESA Space Weather Programme Study Final Presentation Days, ESTEC, Noordwijk, 6-7 December 2001.
4. Pfitzer K.A., Olson W.P. and Mogstad T., *A time dependent source driven magnetospheric magnetic field model*, EOS **69**, 426, 1988.
5. Tada H.Y., *Equivalent Fluence and Relative Damage Coefficient – Tools For Space Solar Cell Degradation Estimate*, IEEE Trans. Nuc. Sci. **20**, 234, 1973.
6. Lundstedt H., Wintoft P., Hasanov T., Boberg F., Gleisner H. and Kronfeldt I., *Forecasting Space Weather and Effects using Knowledge Based Systems*, this issue.
7. Vette J.I., *The AE-8 Trapped Electron Model Environment*, NSSDC/WDC-A-R&S 91-24, 1991.
8. Sawyer D.M. and Vette J.I., *AP-8 Trapped Proton Environment for Solar Maximum and Solar Minimum*, NSSDC/WDC-A-R&S 76-06, 1976.
9. Meffert J.D. and Gussenhoven M.S., *CRRESPRO Documentation*, PL-TR-94-2218, Environmental Research Papers 1158, Phillips Laboratory, 1994.
10. Hedin A.E., Extension of the MSIS Thermospheric Model into the Middle and Lower Atmosphere, J. Geophys. Res. **96**, 1159, 1991.
11. Nieminen P., *Future ESA R&D Activities*, Round Table on 21st Century Monte Carlo Methods for Space Applications, ESTEC, Noordwijk, 13-15 June 2001.
12. Greant Z., Merrall G., Wilson L.T. and Michlitsch B., *PHP Functions Essential Reference*, Hungry Minds, ISBN 0-0721-2731-7, January 2001.
13. Nebel E. and Masinter L., *Form-based File Upload in HTML*, Internet Engineering Task Force, RFC 1867, November 1995.
14. Adams J.H., *Cosmic Ray Effects on MicroElectronics, Part IV*, NRL Memorandum Report 5901, 1986.
15. Taylor E.A., *Space Weather Market Analysis*, Astrium Ltd., ESTEC contract n° 14069/99/NL/SB, November 2000.
16. St. Laurent S., Johnston J. and Dumbill E., *Programming Web Services with XML-RPC*, O'Reilly Inc., ISBN 0-596-00119-3, June 2001.
17. Mitra N., *SOAP Version 1.2 Part 0: Primer*, World Wide Web Consortium, Working Draft, December 2001.
18. Ibbotson J., *XML Protocol Usage Scenarios*, World Wide Web Consortium, Working Draft, December 2001.
19. Levinson E. and Clark J., *Message/External-Body Content-ID Access Type*, Internet Engineering Task Force, RFC 1873, December 1995.
20. Borenstein N. and Freed N., *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*, Internet Engineering Task Force, RFC 2045, November 1996.
21. Berners-Lee T., Fielding R., Irvine U.C. and Masinter L., *Uniform Resource Identifiers (URI): Generic Syntax*, Internet Engineering Task Force, RFC 2396, August 1998.
22. Kruglanski M., Heynderickx D., Boscher D. and Bühler P., *Time Dependent Radiation-Belt Space Weather Modelling*, Space Environments and Effects Final Presentation Days, ESTEC, Noordwijk, 14-15 December 1999.

6. LIST OF ACRONYMS

ACE	Advanced Composition Explorer
AFRL	Air Force Research Laboratory
ASCII	American Standard Code for Information Interchange
ESA	European Space Agency
ESWS	ESA Space Weather Study
FTP	File transfer Protocol
GOES	Geostationary Operational Earth Satellite
HTTP	Hypertext Transfer Protocol
LET	Linear Energy Transfer
MIME	Multipurpose Internet Mail Extensions
NASA	National Aeronautics and Space Administration
NOAA	National Oceanic and Atmospheric Administration
PHP	PHP: Hypertext Pre-processor
REAT	Radiation Effects Analysis Tools
RPC	Remote Procedure Call
SMTP	Simple Mail Transfer Protocol
SOAP	Simple Object Access Protocol

SPENVIS	Space Environment Information System	WWW	World Wide Web
URL	Universal Resource Locator	XML	Extended Mark-up Language