**Auto-Generation of Climate Output Analysis Script Templates**  
*By Sam & Co.*

**Abstract**

To develop a system capable of automatically interfacing legacy analysis scripts with varying input/output formats with a standard web interface, providing direct and intuitive access to analysis scripts and their output in visual format.

*[QUESTION: Keep general or start of describing GFDL situation?]*

**Problem Statement**

In climate simulation, after a model is run, to analyze the data a climate scientist usually writes a script that manipulates the data and produces output that can be in the form of visual data such as plots, graphs, image comparisons, animations, etc. These analysis scripts often do not follow any standard format for input and output interfacing, and the dependencies of the script vary also. An analysis script utilizes as input either hard-coded values or more usually a set of high-level options such as the start and end dates, model data used, variable name, etc, and lower-level options such as file locations, model data chunk lengths (how a model output data is divided up into multiple files, by year), *[MORE EXAMPLES]*.

With just a few scripts, it is possible for scientists to share the actual files through personal means, describing the process of using the script in-person; however, as hundreds of these analysis scripts have now been written it starts to become necessary to implement a system that provide an intuitive and standardized interface to access and run an analysis script of choice on internal model data output.

**Determining Analysis Inputs from Standard Inputs of Web Interface**

A web interface provides high-level options as well as provides ids for the model and variables, which links to an SQL database. All the information needed for these analysis scripts can be determined from metadata stored in both the SQL Database as well as the actual files themselves.

For the majority of the inputs, the actual functions to extract the data for all the scripts are known and available, but the method of determining which set of functions and in what order (ex. to get C from A, you need f1(A)=B, f2(B)=C) is needed.

Many of these scripts have similar templates, as they follow some underlying standard and/or were written by the same climate scientist. The direct solution to accessing all the analysis scripts is to build a template library that hard-code the methods needed to run each analysis script. While this will work for a small set of analysis scripts, such as used by one lab, in the case of merging multiple labs under an institution, or between major institutions, where the number of scripts becomes too time-consuming and obfuscated to hardcode, a system to automatically determine the template format and values is needed.

**Similar Situations**

Many similar situations arise in other environments, such as *[Spencer’s description of space trajectory script things]*, and *[Rocky’s work/intern thing]*.

- Live Access Server @ GFDL (Does this with public data, and basic hard-coded scripts)
[Proof of concept Live Access Server, limitations thereof]

At GFDL [Describe GFDL], the Live Access Server has been developed which provides a web interface to public model output data, and runs a set of basic hard-coded scripts. It provides a clean and intuitive visual format to the analysis output.

The ability to run extensive scripts in a private and public environment and be provided the output of the scripts in a standardized format would increase the number of tools available to a climate scientist.

THE PROBLEM WITH USING FRE

At GFDL, an system called FRE is used to standardize the process of creating, running, post-processing, and analyzing the climate models run internally (CM2.x, ESMSM etc.). This system is built to work with internal model output and requires physical access to all files and output directories.

DISCUSSION

There have been different solutions and methods used to combat this problem,

- Spencer space stuff solution
- Rocky work/intern stuff solution
- CSDMS
- AutoMake
- Datalog
- S - Statistical Language
- Deductive Synthesis (Space)

METADATA HARVESTING FROM LEGACY MODEL OUTPUT

For new model runs, all the information provided is stored in metadata on and SQL database. However, for older models, a lot of the information can only be found in the actual output files themselves, whose locations are known

- Sql vs Prolog
- Tie in Curator?

DIAGRAMS TO INTERSPERSE

- Data Flow Diagram

![Data Flow Diagram](image)

Figure 1 [Still learning GraphViz]

- Architecture Diagram
- ? -> OWL -> RDF -> XML, levels of logic available

SUCCESS SCENARIO

- A method to automatically determine how to convert standard web inputs to any analysis script input and visualizes the output via web.

BACKGROUND:

- Climate science
- Simulation process

FORTRAN SCRIPTS

- [Why & How]
- Legacy code & Fortran analysis script usage
**MDBI – MODEL DATABASE INTERFACE**
The Model Database Interface (MDBI) provides a web interface to browse all the experiments in GFDL. Furthermore, you can view corresponding experiment information. Users can also compare experiments, generate analysis figures, generate RTS XML and check job status. [MORE INFO]

[Question: This is internal only, but we’re trying to make it available publicly w/ public data, include this or not?]

**FRE - FMS RUNTIME ENVIRONMENT**

The FMS Runtime Environment (FRE) is a toolset for managing experiments from start to finish. This includes tasks such as acquiring source code, compilation, launching jobs to run models, and post-processing the output. [MORE INFO]

- [Picture of Ferret Fremetar process]

![Figure 2 - Current Analysis Process](image)

**FMS – FLEXIBLE MODELING SYSTEM**

FMS is a software framework for supporting the efficient development, construction, execution, and scientific interpretation of atmospheric, oceanic, and climate system models. [MORE INFO]

---

1 [http://www.gfdl.noaa.gov/fms-fre4-usage](http://www.gfdl.noaa.gov/fms-fre4-usage)
2 [http://www.gfdl.noaa.gov/fms](http://www.gfdl.noaa.gov/fms)