"Physics of Auroral Phenomena", Proc. XXXI Annual Seminar, Apatity, pp. 157-159, 2008

Polar Geophysical Institute

MODEL INTEGRATION IN THE FRAMEWORK ATMOSPHERE MODEL (FRAM)

O.V. Martynenko, M.A. Knyazeva *(Murmansk State Technical University)*

Abstract. This work continues the description of the Framework Atmosphere Model (FrAM), which is being developed on the basis of the global Upper Atmosphere Model (UAM), for the research of interrelation of the broad range of various processes and the phenomena in the upper atmosphere. Our previous publications reported about a high-level architecture of the FrAM as an open framework, consisting of the controlling Model Manager and the set of independent Models of separate atmospheric regions and processes, and about the FrAM data structure. Using unified model interface the Model Manager organizes the information exchange of the connected Models and controls the execution of the modeling process according to the task configuration prescribed by a user. Now we describe the method how to connect new independent Model to the framework system and include it into the modeling calculation.

1. The FrAM system architecture

Logical structure of the FrAM (with "default" set of the Models) is presented in Fig. 1.

The state of the modeling object (Earth upper atmosphere) during model calculation has being kept in the computer RAM as multi-dimensional arrays of the values of simulated physical parameters in the spatial grid nodes. These arrays (with attaching of some additional information about array structure and meaning of its content) are referenced below as Datasets. The modeling simulation process in essence is the consecutive changing of the Datasets content in such a way as in every moment they contain the instant snapshot of the modeling object.

The modeling calculation itself is being executed by the Models connected to the FrAM. Functionally each Model represents a method of obtaining the values of physical parameters in the spatial grid nodes. No one external object knows how the Model obtains these values – specific method is encapsulated in the Model. It can be the solving of some first-principles-based physics equations (as in theoretical models), or retrieving of values from database, or usage of some analytical synthesizing as series of polynomial, trigonometric or spherical functions (as in empirical models based on statistical generalization of experimental data), or even direct measurements of these parameters (as in real time systems). External objects only have information about physical meaning of this parameter (what is it) and spatial and temporal point that is attributed to (where and when is it). Thus every Model has being used by the another system elements as "black box" described only in the "input – output" terms.

The FrAM system is an open framework. It means there is a possibility to integrate additional Model blocks, which realise an alternative method of obtaining parameter' numerical values (e.g. first-principles-based models instead of statistical ones) or calculate other physical parameters of the atmosphere.

Models can exchange data through the Manager using the standardized interface. The Manager organizes this data exchange: provides every Model all necessary data from other Models and receives new data calculated by the Model in order to transfer it to other Models. Every Model can use values of parameters provided by other Models.

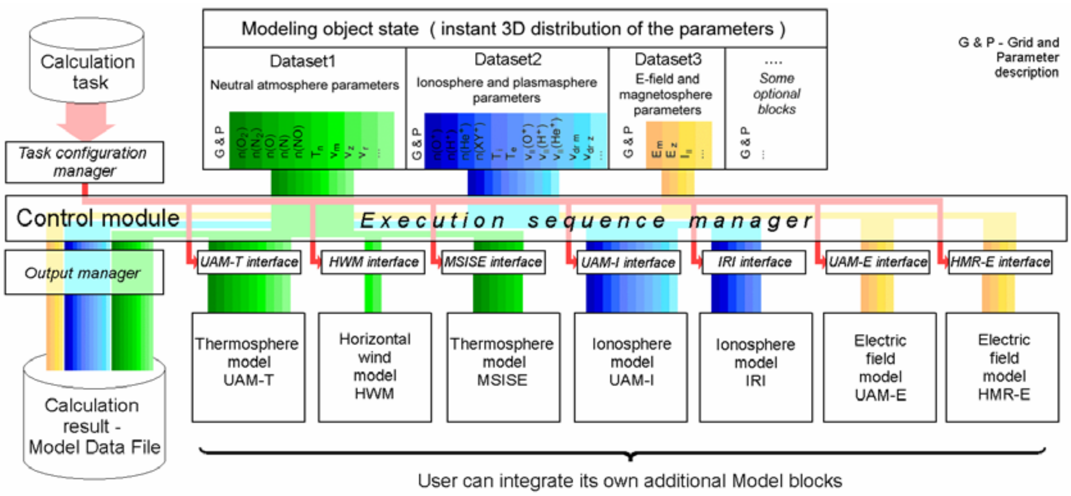

Fig. 1. Logical structure of the FrAM system (including "default" set of the Models)

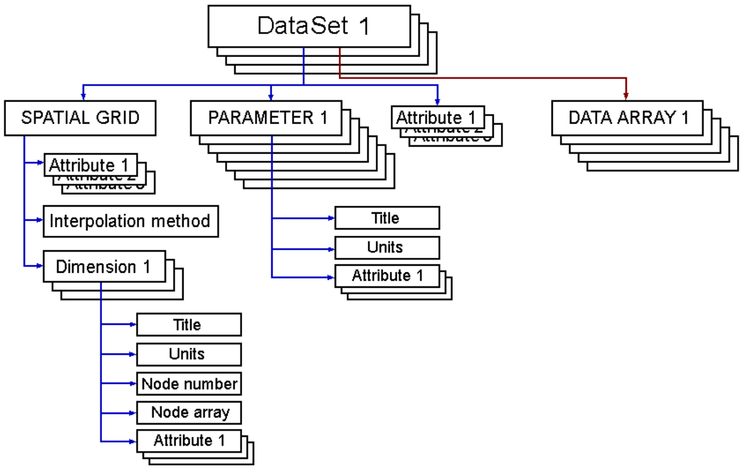

Fig. 2. The Dataset object structure

The key structure of the FrAM model interface is the Dataset object (Fig. 2) that was described in detail in our previous article. The universal form of data exchange between the Models is a passing of Datasets. The Model fills its internal arrays by return values of calculated parameters according to its own method and data structure. But the Model interface block can control which of these parameters to pass to other Models. This separation is realized by using external Dataset and selective passing of data to it. It allows the usage in modeling run of alternative Models of the same processes and regions (with the same physical parameters) in any possible combination in order to switch on/off some physical interrelations and feedbacks.

The Dataset (DS) may be classified according with its relation to the Model as input (iDS) or output (oDS) one (some DS may be iDS and oDS simultaneously). All DS's during model calculation have being kept in the computer RAM, but some of them may be stored additionally in the disk file (stored DS – stDS). This DS may keep initial conditions for the modeling simulation, and they are initialized before the calculation start, or they may keep results of the simulation to be processed lately. Non-stored Datasets exist only during calculation and contain intermediate information.

Hence the Dataset structure is the universal language of data exchange in the FrAM system, and the Model interface modules are the interpreters between this language and the internal data structure of independently developed Models with an additional function – censoring of the Model output according to modeling task configuration.

Another possible problem of inter-Model communication is the spatial grid difference. The FrAM system includes special auxiliary modules for data interpolation (or extrapolation). Formally these modules are designed as other Models – they also return the Datasets – arrays of parameter values in other grid nodes. As these modules are part of the FrAM system, their internal data structure is based on the same Datasets, and they do not need additional interface modules-interpreters.

The Models' execution order can be arbitrary. The user sets it in the task configuration stage, and during the model run stage the Manager executes Models according to this order.

2. Connecting the Model to the FrAM

Due to the FrAM architecture (the "black box" paradigm and cooperation of the Models only through the mediation of the Manager) the connecting procedure in essence is the tuning of the Model-Manager interface.

The FrAM controlling structure (the Metamodel) includes the program code (Model Manager – MM) and the data storage (Metamodel Registry – MR).

The Metamodel Registry keeps information about every registered Model (RM): its designation and list of Model Parameters (MP) with dividing on input (iMP) and output (oMP) ones. Another Registry section stores full list of parameters (RP), which the FrAM "knows" and can calculate:

$$\{RP\} = \bigcup_{\{M\}} \{MP\} \cdot$$

The new Model integration process includes a few steps:

1. Creation of the interface module
   On this step the Model "learns" how to receive and return data organized as Datasets. Technically this step consists in writing of interface subroutine. This subroutine translates input parameters from Dataset structure into internal data structure of a new model, calls the Model subroutine and after its completion re-translates all output parameters into Datasets.
2. Model registration
   On this step the Model "introduces" itself to the Metamodel. After it the Metamodel has the complete information about the Model input and output parameters: what this Model can calculate and what it needs in order to do it. The Metamodel stores this information in the Registry

and in future uses it on the Task configuration stage.

Formally, on this stage the MM creates in the MR new entry for the Model to be integrated and fills it's iMP's and oMP's. If some MP is absent in the RP list, the MM adds it as well.

After completion of these procedures the new registered Model is ready for usage in the FrAM system.

3. Configuration of the modeling task

FrAM system is a flexible one. It provides a user with a set of alternative methods of atmosphere parameters simulation. But in order to use the FrAM a user must define an unambiguous algorithm of the modeling process. It means he must select and assign a specific Model for every modeled parameter and organize the co-operative work of these selected Models. This process consists of next steps:

1. Selection (from RP's) of the physical parameters to be simulated (task parameters – TP's):

   $$\{TP\} \subset \{RP\}.$$

2. Definition of spatial area and creation of the spatial node grid where every TP must be calculated.
3. Grouping of the TP's into the output stDS.
4. Assignment for every TP of the calculating Model (TM) (semi-automatically: if there are a few Models whose oMP's include TP in question, a user must select one of them manually)

   $$\{TP\} \Leftrightarrow \{TM\} \subset \{RM\}.$$

   During the Model assignment step a user also defines all necessary specific options for the TM.

5. Check: every TP must be output parameter of the corresponding TM

   $$TP_i \subset \{oMP(TM_i)\}.$$

6. Definition of iDS and oDS for every TM.
7. Setting TM execution order (there can be repeating calls of some Model to calculate different oMP – they are considered as different TM).
8. Definition of necessary initial conditions (input stDS).
9. Assignment of the DS transfer scheme.
10. Check: for every TM all its iMP's must be oMP's of some other TM that works before or must be in input stDS.

    $$\{iMP(TM_i)\} \subset (\bigcup_{j<i} \{oMP(TM_j)\}) \bigcup stDS.$$

    Additional Models are added to the TM list (and execution order and DS transfer scheme) until it will be true.

11. Assignment of inter- and extrapolation modules to provide DS transfer between the TM's with different spatial grids.

These procedures create the new modeling World with complete closed system of inter-relation laws of its elements.

The last stage in the modeling calculation preparation process is the definition of the starting state of the new created World (initial condition), external forcings that influence it, and assignment of output information: what, where and how often should be saved during modeling simulation.

Acknowledgments. This work was supported by the grant № 08-05-98830 of the Russian Foundation for Basic Research.

References

1. Namgaladze A.A., O.V. Martynenko, A.N. Namgaladze. Global Model of the Upper Atmosphere With Variable Latitudinal Integration Step // Geomagnetism and Aeronomy International, V.1, No.1, 53-58, 1998.
2. O.V. Martynenko, M.M. Gladkikh, I.V. Artamonov, D.V. Sobolev. On usage of the object oriented data structure for geophysical data storage and processing // Physics of Auroral Phenomena: Proceedings of 30th Annual Seminar – Apatity, 2007. – p. 171-173.
3. I.V. Artamonov, M.M. Gladkikh, O.V. Martynenko, A.A. Namgaladze, D.V. Sobolev. Framework Atmosphere Model - software tool for model coupling. *Geophysical Research Abstracts, Vol. 9,* EGU2007-A-10166, 2007
4. O.V. Martynenko, A.A. Namgaladze, M.M. Gladkikh, I.V. Artamonov. Framework Atmosphere Model – modeling environment with already adjusted system of physical interrelations. *Earth: Our Changing Planet. Proceedings of IUGG XXIV General Assembly*. Perugia, Italy, July 2-13, 2007. p.2992