Atmos. Chem. Phys. Discuss., 4, S3524–S3529, 2004
www.atmos-chem-phys.org/acpd/4/S3524/
European Geosciences Union

Atmospheric Chemistry and Physics Discussions

Interactive comment on "Technical Note: The Modular Earth Submodel System (MESSy) – a new approach towards Earth System Modeling" *by* P. Jöckel et al.

P. Jöckel et al.

Received and published: 1 February 2005

We thank the two anonymous referees for their valuable, clear, and constructive comments. In the following, we respond to their comments:

1. Abstract: Indeed, the abstract needs to be clarified. MESSy is neither only an interface, nor is it (yet) an Earth System Model (ESM) on its own. But what is it exactly? MESSy is simply a 'way of doing it', and comprises three parts:
   - a standard interface to couple together processes coded as 'submodels'
   - a comprehensive, flexible, and extendable set of submodels

Full Screen / Esc
Print Version
Interactive Discussion
Discussion Paper
EGU

- a coding standard, which is mainly based on the Fortran95 standard

The model system has so far been tested with the General Circulation Model (GCM) ECHAM5 (see supplement), a number of boxmodels for various processes, and the GCM ECHO-G (project EGMAM, http://wekuw.met.fu-berlin.de/WWKE/index.php?loop=projects?=ger&page=Projekte/details.php&proj_name=EGMAM). More information can be found on the MESSy web pages (http://www.messy-interface.org). In general, MESSy can be linked to any model, either global or regional, GCM or Chemistry Transport Model (CTM), atmosphere, ocean, etc.

We point out that MESSy is not a coupler in the classical sense, i.e., an independent stand-alone program organizing the communication between other programs. Nevertheless, such approaches can be combined. For instance, an atmospheric model written according to the MESSy standard can easily be coupled to an ocean model by means of an external coupler.

However, our vision is indeed to ultimately form a complete ESM by a (large) set of submodels, and a base model which contains not much more than a central clock and runtime control. The advantage of this approach is obvious. One can start from an existing model (e.g. a GCM) and re-implement it step by step, i.e., process by process. This ensures the availability of a 'state-of-the-art' model for scientific applications at any time of the development, since development of submodels and application can be performed in parallel. Moreover, the step-by-step approach guarantees robust results; in many cases the re-implementation can be performed to give numerical identical results.

In the revised version these points are mentioned in the abstract.

2. Objectives: We agree with referee #1 that the formulation of the objectives might have been misleading. The list of 'objectives / requirements' appears to be a wish-list with conditions, which indeed applies to any kind of ESM, whereas the

ACPD

4, S3524–S3529, 2004

list of 'prerequisites' shows one method (possibly among others) of achieving this. To clarify this, we reformulated the text.

3. Disadvantages: We agree with referee #1 that potential disadvantages need to be discussed (In the revised version we introduced a new section 'Discussion'):

   - Systematic performance tests are difficult, since fair tests would require a likewise comprehensive and flexible system coded in some 'old-fashioned' way. Since data exchange in the MESSy structure is performed within one executable, we expect the performance to be better than that of systems of several specific executables communicating via an additional external coupler.
   - A similar argument holds for the memory management: Applying the 'classical' coupler-approach, exchanged data need to be stored in both specific executables communicating to each other, and in addition in the coupler (at least temporarily). In contrast, data need to be stored only once in the MESSy approach. All processes (submodels) have access to the same consistent dataset. This is achieved by a pointer-arithmetic based memory-management with minimum overhead.
   - Of course 'quick and dirty' testing is still possible within MESSy, since these tests are mostly related to a specific submodel and therefore independent of the interface. However, we emphasize that these kind of tests are only reasonable during the development and debugging-phase and should not remain in the code for production runs (see also the preamble of our supplement).

4. Specific comments of referee #1:

   (a) Online-couplings: We agree and changed the text.

ACPD

4, S3524–S3529, 2004

(b) Quantification of coupling processes: We agree that the quantification of coupling processes is very difficult, and indeed local budgeting is not sufficient. However, MESSy offers new possibilities to systematically address the issue of feedback mechanisms, since the data flow is much more transparent and can be followed much easier as in many other state-of-the-art models. This is a necessary but not a sufficient condition. The text has been changed in the revised version.

(c) requirements: We changed the text accordingly.

(d) interface: We mention this in the abstract now.

(e) Figure 6: Indeed this needs to be clarified. The submodel core layer, as indicated by the separated box in Figure 6, is independent of the base model. At the final development state of MESSy, also the second layer from below (submodel interface layer (SMIL), pale yellow area) is likewise independent of the base model, since it solely makes use of the MESSy infrastructure, for instance the data transfer and export interface. Last but not least, the base model dependence of the second layer from above (base model interface layer, BMIL) is minimized by extensive usage of the MESSy infrastructure. Only well defined connections (as indicated by arrows between BML and BMIL in Figure 4) are required.

(f) ECHAM5/MESSy: Taken literally, ECHAM5 has neither been implemented into MESSy nor vice versa. ECHAM5 as a base model has been modified to be conform to the MESSy interface (see Appendix), and some processes of ECHAM5 have already been reimplemented as MESSy submodels. The final aim is to have all ECHAM5 processes completely modularized as MESSy submodels. In the revised version we write 'connected to'.

5. References: We agree with the referee that references of peer-reviewed articles would be better than references to web-pages or 'grey literature' (as for instance

ACPD

4, S3524–S3529, 2004

internal reports). However, in those cases where we referenced web-pages or 'grey literature', adequate peer-reviewed publications are not available. Peer reviewed articles about technical details of modeling and implementation techniques are unfortunately rare in this field. From our point of view this reflects the lack of appreciation of code development in the scientific community. Code development is generally not regarded as science (contrary to instrument development, for which specialized journals are available).

Furthermore, referencing some arbitrary modeling studies in our introduction would be misleading. First, most of these studies apply to classical GCMs rather than ESMs. Second, we do not provide a comprehensive overview. This would be beyond the scope of our manuscript. On the other hand, our introduction is a very critical review, listing some severe problems. If we give references here, we would un-justifiably cast a negative light on some selected models.

Nevertheless, we found a very recent publication describing some of these issues in a wider frame, and added the reference.

6. Acronyms: Since the manuscript is a 'Technical Note' it is occasionally unavoidable to be somehow 'technical' for the sake of precision. Nevertheless we make sure that all acronyms are written out the first time they are used.

7. Vector and scalar architectures: Due to the strict modularization, MESSy allows for a flexible handling on both, vector- and scalar- architectures. Compiler capabilities like automatic vectorization (vector blocking) and inlining can be applied straightforwardly. Last but not least, for a super-optimization, vector- and scalar-code of the same process can coexist and switched on/off depending on the architecture used.

8. Multi-developer issue: This is a good point. Of course, the implementation of a MESSy structure by several developers can be organized by using software tools

ACPD

4, S3524–S3529, 2004

like CVS (or other systems). This is however not directly related to the MESSy concept and we prefer to leave it out.

Interactive comment on Atmos. Chem. Phys. Discuss., 4, 7139, 2004.