

SpringerBriefs in Earth System Sciences

Series Editors

Kevin Hamilton

Gerrit Lohmann

Lawrence A. Mysak

For further volumes:

<http://www.springer.com/series/10032>

Sophie Valcke · René Redler
Reinhard Budich

Earth System Modelling – Volume 3

Coupling Software and Strategies

 Springer

Dr. Sophie Valcke
CERFACS, Av. Coriolis 42
31057 Toulouse Cedex 01
France
e-mail: valcke@cerfacs.fr

Reinhard Budich
Max-Planck-Institut für Meteorologie
Bundesstraße 53
20146 Hamburg
Germany
e-mail: reinhard.budich@zmaw.de

Dr. René Redler
Max-Planck-Institut für Meteorologie
Bundesstraße 53
20146 Hamburg
Germany
e-mail: rene.redler@zmaw.de

ISSN 2191-589X
ISBN 978-3-642-23359-3
DOI 10.1007/978-3-642-23360-9
Springer Heidelberg Dordrecht London New York

e-ISSN 2191-5903
e-ISBN 978-3-642-23360-9

Library of Congress Control Number: 2011938153

© The Author(s) 2012

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

*To all people who work hard every day to
make the world a better place*

Preface

Climate modelling in former times mostly covered the physical processes in the atmosphere. Nowadays, there is a general agreement that not only physical, but also chemical, biological and, in the near future, economical and sociological—the so-called anthropogenic—processes have to be taken into account on the way towards comprehensive Earth system models. Furthermore these models include the oceans, the land surfaces and, so far to a lesser extent, the Earth’s mantle. Between all these components feedback processes have to be described and simulated.

Today, a hierarchy of models exist for Earth system modelling. The spectrum reaches from conceptual models—back of the envelope calculations—over box-, process- or column-models, further to Earth system models of intermediate complexity and finally to comprehensive global circulation models of high resolution in space and time. Since the underlying mathematical equations in most cases do not have an analytical solution, they have to be solved numerically. This is only possible by applying sophisticated software tools, which increase in complexity from the simple to the more comprehensive models.

With this series of briefs on “Earth System Modelling” at hand we focus on Earth system models of high complexity. These models need to be designed, assembled, executed, evaluated, and described, both in the processes they depict as well as in the results the experiments carried out with them produce. These models are conceptually assembled in a hierarchy of sub models, where process models are linked together to form one component of the Earth system (Atmosphere, Ocean, ...), and these components are then coupled together to Earth system models in different levels of completeness. The software packages of many process models comprise a few to many thousand lines of code, which results in a high complexity of the task to develop, optimise, maintain and apply these packages, when assembled to more or less complete Earth system models.

Running these models is an expensive business. Due to their complexity and the requirements w.r.t. the ratios of resolution versus extent in time and space, most of these models can only be executed on high performance computers, commonly called supercomputers. Even on todays supercomputers typical model experiments

take months to conclude. This makes it highly attractive to increase the efficiency of the codes. On the other hand the lifetime of the codes exceeds the typical lifetime of computing systems, including the specifics of their architecture, roughly by a factor of 3. This means that the codes need not only be portable, but also constantly adapted to emerging computing technology. Whereas in former times computing power of single processors—and such of clustered computers—was resulting mainly from increasing clock speeds of the CPUs, today's increases are only exploitable when the application programmer can make best use of the increasing parallelity off-core, on-core and in threads per core. This puts additional complexity in areas like IO performance, communication between cores or load balancing to the assignment at hand.

All these requirements put high demands on the programmers to apply software development techniques to the code making it readable, flexible, well structured, portable and reusable, but most of all capable in terms of performance. Fortunately these requirements match very well an observation from many research centres: due to the typical structure of the staff of the research centres code development oftentimes has to be done by scientific experts, who typically are not computing or software development experts. So code they deliver deserves a certain quality control to assure fulfilment of the requirements mentioned above. This quality assurance has to be carried out by staff with profound knowledge and experience in scientific software development and a mixed background from computing and science.

Since such experts are rare, an approach to ensure high code quality is the introduction of common software infrastructures or frameworks. These entities attempt to deal with the problem by providing certain standards in terms of coding and interfaces, data formats and source management structures etc., that enable the code developers as much as the experimenters to deal with their Earth system models in a well acquainted, efficient way. The frameworks foster the exchange of codes between research institutions, the model inter-comparison projects so valuable for model development, and the flexibility of the scientists when moving from one institution to another, which is commonplace behaviour these days.

With an increasing awareness about the complexity of these various aspects scientific programming has emerged as a rather new discipline in the field of Earth system modelling. Coevally new journals are launched providing platforms to exchange new ideas and concepts in this field. Up to now we are not aware of any text book addressing this field, tailored to the specific problems the researcher is confronted with. To start a first initiative in this direction, we have compiled a series of six volumes each dedicated to a specific topic the researcher is confronted with when approaching Earth System Modelling:

1. Recent developments and projects
2. Algorithms, code infrastructure and optimisation
3. Coupling software and strategies
4. IO and postprocessing
5. Tools for configuring, building and running models
6. ESM data archives in the times of the grid

This series aims at bridging the gap between IT solutions and Earth system science. The topics covered provide insight into state-of-the-art software solutions and in particular address coupling software and strategies in regional and global models, coupling infrastructure and data management, strategies and tools for pre- and post-processing, and techniques to improve the model performance.

Volume 1 familiarizes the reader with the general frameworks and different approaches for assembling Earth system models. Volume 2 highlights major aspects of design issues that are related to the software development, its maintenance and performance. Volume 3 provides an overview and introduction into the major coupling software products developed and used in the climate modelling community. The different technical attempts or possibilities are discussed from the software point of view to solve the coupled problem. Once the coupled model is running data are produced and postprocessed (Volume 4). The whole process of creating the software, running the model and processing the output is assembled into a workflow (Volume 5). Volume 6 describes coordinated approaches to archive and retrieve data.

Hamburg, June 2011

Reinhard Budich
René Redler

Acknowledgments

We would like to thank all authors for their tenacity in the writing of this Volume, and Michel Desgagné and Anthony Thevenin for their very constructive review.

Contents

1	Introduction	1
2	TDT: A Library for Typed Data Transfer	3
2.1	Introduction	3
2.2	Architectural Overview	4
2.3	Benefits and Limitations of TDT	5
2.4	Model Coupling with TDT	7
2.5	TDT in Practice	8
2.5.1	The Data Description File	8
2.5.2	The Configuration File	9
2.5.3	Implementation in the Original Codes	10
2.6	Additional Technical Details	11
2.7	Conclusions and Perspectives	11
3	The Model Coupling Toolkit	13
3.1	Introduction	13
3.2	Architectural Overview and Programming Philosophy	14
3.3	MCT Datatypes and Methods	16
3.4	MCT Multi-Language Interface	19
3.5	Conclusions and Perspectives	20
	References	21
4	The OASIS Coupler	23
4.1	Introduction	23
4.2	Architectural Overview	24
4.3	Coupling Configuration	25
4.4	Process Management	25
4.5	Communication: The OASIS PSMILe Library	25
4.6	Coupling Field Transformation and Regridding	27

4.6.1	Transformation and Regridding in OASIS3	27
4.6.2	Transformation and Regridding in OASIS4	28
4.7	Performances	29
4.8	User Community	30
4.9	Conclusions and Perspectives	31
	References	32
5	The Flexible Modeling System	33
5.1	Introduction: The Emergence of Modeling Frameworks	33
5.2	Architectural Overview	34
5.3	Physical Architecture of FMS Coupled System	36
5.4	The Exchange Grid	38
5.5	Data Assimilation	40
5.6	Conclusions and Perspectives	41
	References	41
6	The Earth System Modeling Framework	43
6.1	Introduction	43
6.2	Architectural Overview	44
6.3	Components in ESMF	46
6.4	Remapping in ESMF	47
6.5	Adopting ESMF	47
6.5.1	Wrapping User Code in ESMF Components	48
6.5.2	Adapting Data Structures	48
6.5.3	Registering User Methods	49
6.5.4	Coupling Between ESMF Components	51
6.5.5	Executing the Application	52
6.6	Alternative Forms of Coupling	52
6.7	Conclusions and Perspective	53
	References	53
7	The Bespoke Framework Generator	55
7.1	Introduction	55
7.2	Architectural Overview	56
7.3	BFG1	57
7.4	BFG2	59
7.4.1	Argument Passing	60
7.4.2	Subroutine Entry Points	61
7.4.3	Scientific API	61
7.4.4	Initialisation	62
7.4.5	Control	62
7.4.6	XSLT Implementation	62
7.4.7	Frameworks as Targets	63
7.4.8	Grids	63
7.4.9	The GENIE ESM—An Example	64

Contents	xv
7.5 Related Systems	64
7.6 Conclusions and Perspectives	65
References	67
8 Future Perspectives	69
Glossary	71
Index	75

Contributors

V. Balaji Princeton University, New Jersey, USA, e-mail: balaji@princeton.edu

Cecelia DeLuca NOAA/CIRES, Boulder, CO, USA, e-mail: cecelia.deluca@noaa.gov

Rupert Ford University of Manchester, Manchester, UK

Robert Jacob Argonne National Laboratory, Argonne, USA, e-mail: jacob@mnscl.anl.gov

Jay Larson Research School of Computer Science, The Australian National University, Canberra, Australia, e-mail: larson@mnscl.anl.gov

Ciaron Linstead Potsdam Institute for Climate Impact Research, Potsdam, Germany, e-mail: linstead@pik-potsdam.de

René Redler Max-Planck-Institut für Meteorologie, Bundesstraße 53, 20146, Hamburg, Germany, e-mail: rene.redler@zmaw.de

Graham Riley University of Manchester, Manchester, UK, e-mail: graham.riley@manchester.ac.uk

Gerhard Theurich Science Applications International Corporation, McLean, VA, USA,

Sophie Valcke CERFACS, Av. Coriolis 42, 31057, Toulouse Cedex 01, France, e-mail: sophie.valcke@cerfacs.fr