Environmental Modelling & Software 26 (2011) 1747–1758

Contents lists available at ScienceDirect

Environmental Modelling & Software

journal homepage: www.elsevier.com/locate/envsoft

On the use of a high-performance framework for efficient model coupling in hydroinformatics

N. Malleron [a,*], F. Zaoui [b], N. Goutal [a,b], T. Morel [c]

[a] LHSV c/o EDF – R&D, 6 quai Watier, 78401 Chatou cedex, France
[b] EDF – R&D, LNHE, 6 quai Watier, 78401 Chatou cedex, France
[c] CERFACS, 42 Avenue Gaspard Coriolis, 31057 Toulouse Cedex 01, France

ARTICLE INFO

Article history:
Received 5 October 2010
Received in revised form
15 May 2011
Accepted 27 May 2011
Available online 18 July 2011

Keywords:
Model coupling
Framework
Hydraulic engineering
Free surface flow

ABSTRACT

Many situations in hydraulic engineering require the use of numerical modelling. Developments in this field began about 50 years ago and the resulting computational codes have now reached maturity. A current trend (not restricted to hydraulics) consists in global modelling and building Decision Support Systems. This necessitates taking into account a large number of phenomena and often presupposes coupling appropriate computational codes. Moreover, many models have been built in recent years but this requires much time and effort since whole rivers (and sometimes their tributaries) are often under consideration. So, when undertaking a new study, the reuse of existing models considerably reduces engineering time. The possible lack of information in an existing model can be compensated for by coupling it to another model, thereby giving access to the missing information. This study investigates the available techniques for coupling hydraulic models with different spatial dimensions (1D and 2D). An easy-to-use framework, called PALM, is compared to the other existing tools, and adopted for further study. The way to perform coupling of existing and relatively complex code in such a framework is detailed. We focus on the reduced invasivity of PALM, which makes it very convenient for algorithm testing and prototyping. Results on real-world hydraulic cases are shown, in order to demonstrate the performance of this approach, compared to less accurate coupling, based on file exchange for example.

Software and availability

Name of Software: MASCARET
Contact address: Nicole Goutal, 6 quai Watier, 78401 Chatou Cedex, FRANCE, nicole.goutal@edf.fr
Availability: Freeware, upon request. An Open Source version will be available soon
Program language: Fortran90
Name of Software: TELEMAC2D
Contact address: SOGREAH, 6 rue de Lorraine, 38130 Echirolles, FRANCE, telemac@sogreah.fr
Software requirements: Perl, MPI2 (MPI library) + METIS for parallel computing
Availability: Open Source available on the website http://www.telemacsystem.com
Program language: Fortran90
Name of Software: PALM
Contact address: The Palm Team at Cerfacs, 42 av. Gaspard Coriolis, 31100 Toulouse, FRANCE
Software requirements: A compatible Fortran/C compilers suite (pgi and gnu compilers have been used by the authors).
Availability: Open Source upon request on the website: http://www.cerfacs.fr/globc/PALM_WEB/index.html
Program language: C++; Tcl/Tk

1. Background and motivation

In many applications, using a single computational code to deal with a specific type of phenomenon may be inadequate. Decision Support Systems (DSS) usually need global system modelling, involving a significant number of phenomena. The physics driving these phenomena can be different: Agricultural management systems (used to build a global farming model) (Holzworth et al., 2010) are built using at least crop, soil and weather models; Meteorological applications (Warner et al., 2008) are built using interactions between oceans and atmosphere models, *etc*. An

* Corresponding author. Tel.: +33130877881; fax: +33130878109.
E-mail address: nicolas.malleron@edf.fr (N. Malleron).

1364-8152/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.envsoft.2011.05.017

example of a DSS based on an impressive number of coupled environmental models was built by Lam et al. (2004).

However, it is not in the interest of either developers, or users, to give up the mono-physic or mono-dimensional tools developed previously. The latter are often the result of long and painstaking research and developments and have already proved their robustness, accuracy and efficiency. Many other arguments favour the reutilisation of existing computational codes. Users often have little time to perform their studies in an engineering context. Consequently, they dislike having to change tools and learning to use new ones. Furthermore, these tools are sometimes accredited and their use can be enforced by a certification or control office.

Coupling allows the reuse of such well-established codes, in order to provide users with reliable and powerful tools covering a wider range of applications and degrees of freedom. It actually tends to be the solution chosen by many development teams. They strive to make their codes interoperable, i.e. capable of communicating and working with other codes, instead of trying to develop phenomenon-specific codes. Leaving such *ex nihilo* modelling to one side, many possibilities remain. Brandmeyer and Karimi (2000) reviewed coupling methodologies and distinguished 5 levels of coupling, listing their respective advantages and drawbacks. The use of a framework appears to be the best choice. It guarantees the independence of codes and avoids damaging integration, since each code has its own logic of development and maintenance, as well as its own semantics.

By revisiting the low-level coupling of the free surface hydraulic computational codes MASCARET and TELEMAC2D, initially developed by Tchamen and Roubtsova (2007) and continued by Malleron et al. (2010), we aim to make these computational codes interoperable. Finally, the advantages of using suitable frameworks to reduce the workload involved in code modification are highlighted.

In section 2, the available solutions for coupling codes are described. The main characteristics of these solutions are compared. The properties of PALM coupler are highlighted. In section 3, this coupler is used to couple the two free surface hydraulic codes MASCARET and TELEMAC. Results from different real-world cases are investigated in section 4: namely the well-known Malpasset dam break (Goutal, 1999) and the case of the 2007 flood event passing through Marckolsheim dam, located on the French part of the Rhine river. Computational times are then discussed before presenting the conclusions and perspectives in section 5.

2. Interoperability and coupling framework

In this study, we focus on component (corresponding to a part of a computational code or an entire computational code itself) frameworks. Service oriented architecture (SOA) is not considered. According to the state-of-the-art, SOA is relevant for coarse grain interoperability and when discoverable services are needed. This is useful in loose coupling. In the case considered here, a great deal of information is exchanged (strong coupling) and a component-based framework is more relevant. For more details on SOA, please see Lewis and Wrage (2004) for an example.

2.1. Comparison of available solutions for an efficient coupling

2.1.1. An overview

Numerous technical solutions to assemble model codes are available:

- The first one consists of merging the codes. This solution is portable and efficient since communication is based on memory exchange. Nevertheless, it is neither easy to implement with existing codes, nor flexible, since coupling algorithms are consequently hard coded.
- Another solution consists of using existing communication protocol such as MPI, CORBA, *etc*. This solution is however not easier to implement than the first one since it requires the developer to be a protocol expert. It is once more not flexible, less adapted to sequential coupling and neither very efficient nor portable.
- The third solution consists of using frameworks or libraries. This solution is flexible, efficient and portable. It also allows the use of generic utilities (parallelisation, regridding, time management, *etc.*) It is probably the best solution in a controlled development environment. However, it requires the respect of some norms during the development of a new product or the significant modification of an existing one.
- The fourth and last solution is the use of a coupler. Like the previous technique, it is flexible, efficient and portable. It also allows the use of generic transformations/regridding and probably represents the best solution to couple independently developed codes.

Jagers (2010) presents a very interesting review of available tools for code coupling (CCA, ESMF, HLA, KEPLER, MCT, OASIS, OMS, OpenMI, TIME) belonging to the third or fourth category listed above. Since they are well-described in Jagers (2010), we will not give any more details about them. We will rather focus on some tools, not mentioned in Jagers (2010) but that are also of interest : SME, YACS, PALM, BFG, TDT.

- Spatial Modelling Environment[1] (SME), transparently links icon-based modelling environments with advanced computing resources. It allows modellers to develop simulations in a graphical environment, not requiring any programming knowledge. Automatic code generator built simulation and allows to distribute processing over a network of parallel and serial computers. Transparent access to state-of-the-art computing technics is provided. The environment imposes the constraints of modularity and hierarchy in model design, and supports archiving of reusable model components defined in Modular Modelling Language (MML).
- PALM (Piacentini, 2003; Buis et al., 2006) was originally developed to deal with data assimilation in the scope of the MERCATOR ocean forecasting project (Bahurel, 2006). The different tasks-running a forecast, applying the observation operator, computing misfits, approximating the error statistics, inverting matrices, minimising a cost function… - were thought to be independent pieces of code to be assembled within a portable, flexible and efficient framework. The PALM Team at CERFACS was in charge of the development of a dynamic parallel coupler and its user interface that can be used on a variety of platforms subject to the efficiency constraints of operational applications.
- YACS (Ribes and Caremoli, 2007) is a module of the Open-Source SALOME (EDF-CEA) framework[2]. YACS is an upgrade of the CALCIUM coupling module (Berthou and Caremoli, 1999) developed in 1993. It is based on a design-by-interface and is quite similar to PALM framework in its use.
- The Bespoke Framework Generator[3] (BFG), developed by the University of Manchester, is not a framework. It is a wrapper code generator which allows to build a coupled model with any

[1] http://www.uvm.edu/giu/SME3.
[2] http://www.SALOME-platform.org/.
[3] http://intranet.cs.man.ac.uk/cnc/projects/bfg.php.