Organization of Code Development for Collaboration with Commercial Partners

Study

Revision 1.2, February 2012.
(Revision 0: Dec. 2010)

Igor Grešovnik

COBIK,
Laboratory for Advanced Materials Systems

Contents:

1 INTRODUCTION

This document contains a proposal of how to organize collaboration between a Commercial partner and the Laboratory for Advanced Materials Systems of the Centre of Excellence for Biosensors, Instrumentation and Process Control (COBIK).

The basic presumptions are the following:

- Commercial partner is an industry leader in its field with long tradition and influential global market position. It is a high tech company that is aware of the meaning of investment in technological development.
- Laboratory for Advanced Materials Systems ("the Laboratory"), lead by Professor Šarler, is a world renowned group for development of complex numerical simulations building on a long term tradition and with top level achievements, including a number of break through developments within its field.
- Commercial partner is interested in using the knowledge accumulated in the Laboratory and in inclusion of this knowledge in the development of its products and services.
- Commercial partner is interested in long term collaboration where the Laboratory would continuously support it with top-end knowledge from the field and with new developments.
- The laboratory is highly interested in such collaboration because it will give the group opportunity to further promote the practice of direct verification of new research achievements in industrial practice. On basis of such collaboration, the laboratory will also be able to expand its research potential and attract more high level experts, which will further improve its position as leading laboratory in the field.
- For Commercial partner it is important that it retains full control over results of such collaboration. This includes the ability to easily use results without having to apply for extensive additional support from the laboratory, to easily establish adherence of products of collaboration with its internal standards, to easily integrate the produced results into its technical development environment, and also to protect its intellectual property and business secrets.
- For the laboratory it is important to retain its ability of good research and academic work, which includes international collaboration and exchange of ideas. This is crucial in order to keep pace with top level global developments in the field and thus to maintain fitness to produce top level results and to offer good support to its industrial partners in long term.

It is important to realize that numerical models developed within the laboratory are very complex and are closely linked to practical implementation in code. It is impossible to separate

theoretical development from a well designed code framework, because theoretical development in numerical fields can only be accomplished within such a framework.

The usual arrangement of the laboratory with its industrial partners is that the laboratory develops the complete numerical code, of which industrial partners are entitled to the code binaries with updates, to certain non-transferable rights to use the code and to certain level of support that is agreed between partners on case-to-case basis. Such an arrangement might be insufficient to satisfy Commercial partner's demands for control over the developed work results, therefore the arrangement proposed herein is a bit different and strives after optimal coverage of interests and needs of both partners. Most importantly, the proposal gives much emphasis on feasibility of the arrangement and to the ability to produce results efficiently.

2 DEVELOPMENT OF SIMULATION CODE

The Laboratory has historically used different numerical codes and its Ph.D. students have often developed their research codes on various different platforms. While such diversity is partially a consequence of large complexity of problems being solved, we are unifying the simulation software development framework in order to enable more efficient development, better collaboration within the team and easier introduction of newcomers into work that requires high level of technical skills.

The basic development platform is MS .NET (or the cross-platform Mono framework) with C# as the primary programming language. Such a choice has been made for variety of reasons, including the following:

- The framework provides an extensive library base.
- Development made on this platform is easily transferred across platforms. Maintenance of cross-platform compatibility is much easier than with native languages (even as compared with pure ANSI C, which is a model environment for development of native cross-platform applications).
- The C# language has a very clean, simple and well elaborated object oriented syntax, which makes such a platform one of the most efficient widely used, actively developed and well supported platforms currently available.
- Despite clear and simple syntax, the capabilities of the C# language go beyond capabilities of many of its competitors. C# is a multi-paradigm programming language, it is reflective and has built-in support for parallel and secure multitier applications that can take full advantage of distributed computing environments (both on homogeneous or inhomogeneous networks).
- Programming in C# is less error prone as compared to other language. This enables involvement of staff members that are not trained IT professionals but can achieve excellent results in development of numerical algorithms.

For the simulation framework, we have defined very clear standards of how to integrate different functionality needed in applications in order to preserve highly modular and cross platform

structure of the basic framework. Our point of view is that we can easily implement similar rules in the case of collaborative development of numeric models for Commercial partner. We think this is the best way to satisfy our interests and make the collaboration feasible, efficient, productive and profitable.

2.1 Structure of Code

The simulation code developed in the Laboratory is arranged in a layered hierarchical structure (Figure 1). The basic level consists of general libraries that are shared between higher level libraries and applications. Requirements for these libraries are rather strict and include the following:

- Base libraries may not depend on libraries that would put any restrictions on licensing of these libraries and products derived from them. One must be able to license any derived software under commercial as well as free open source licenses. Such requirement is crucial for scientific exchange that enables the group to keep pace with state-of-the-art.
- It must be straight forward to transfer the libraries across platforms. As has been stressed before, theoretical development of numerical fields can not be separated from good code base. It depends on flexible software implementation that can be run in different environments, including high performance supercomputers, in order to experiment with ideas and test how different ideas perform in practice.
- Code in the base libraries must be well designed, modular, extensible, easily readable and well documented. Rapid prototyping must be done at designated places elsewhere, and code can be moved into base libraries only when it reaches a certain level of maturity.

On top of base libraries there is a level of other utility libraries that are still considered rather general, but do not satisfy all the requirements for being part of the base libraries collection. This includes e.g. some graphical libraries that don't satisfy the condition of being easily transfer across platforms, or some external libraries used in application that induce licensing restrictions on the derived software.

On top of utility libraries there are different application frameworks. These include e.g. the optimization framework and the mesh free simulation framework. These frameworks are complete enough that it is possible to quickly derive simple applications from them. This quality is utilized by researchers and students who often need to set up fully functional applications for development and testing of new algorithms. The application framework will also include a small set of model applications. These will serve demonstration purposes, as models for building specialized applications, as testing frameworks, as representative and popularization software, etc.

The top level consists of specific customized applications produced specifically for given purpose (e.g. in the scope of some project or on partner's demand). The laboratory has produced several such applications in the past. While we try to include much of our development at levels up

to the general simulation frameworks and reuse it as much as possible, customized application level contains all the code that is case specific or that contains sensitive information from our partners' point of view. In contrary to lower levels of code stack, this level is kept closed and inaccessible to others in order to protect any information that is related to our partners' specific.

The customized applications usually include an integration layer that is used to integrate the application with the existing software environment of our partners.

The key ideas of collaboration proposal are as follows:

- Development of numerical models remains organized in the described manner.
- All code for which Commercial partner does not want to be accessible to others is put into the specific application level. Code here will be accessible only to Commercial partner and specific Laboratory staff members, who will be bound by appropriate non-disclosure agreements.
- We will continue to accumulate the general code related to numerical models at levels of the code stack of which the Laboratory retains all rights and can use them without restrictions. We believe this is also in the interest of Commercial partner because such arrangement will enable us to accumulate in the same code the results of other collaborations and of high level research work generated in different groups, which will ensure efficiency and reduce the cost of development.
- In order to enable Commercial partner to retain sufficient control over the products of collaboration (and also over other knowledge and work that has accumulated in the Laboratory over years), we will enable Commercial partner access to our source code and enable its staff to work on this code. In such a way Commercial partner will also get sufficient knowledge for making full use of the code produced by the laboratory.
- In order to allow the laboratory to perform high quality research work, the laboratory will retain all rights on its code. There will be an additional safeguard for Commercial partner. After a certain period Commercial partner will gain rights to create an independent branch of Laboratory's source code, on which it will have all rights (while the laboratory will at the same time retain all rights on its original branch of code).

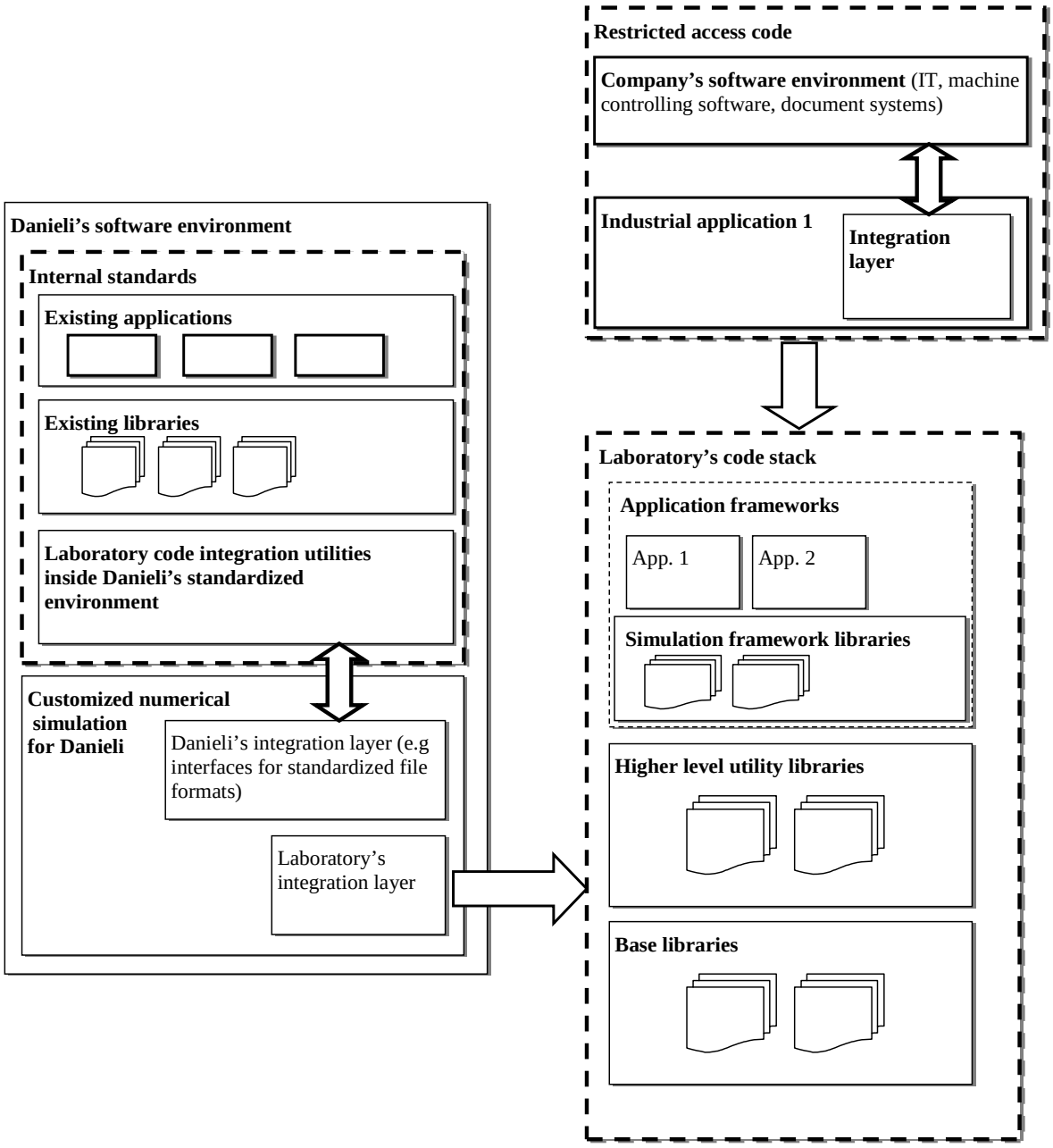

Figure 1: Code stack hierarchy.

References:

[1] Igor Grešovnik: Coordination of software development in COBIK and Laboratory for Multiphase Processes. Treatiese, COBIK, 2011.

[2] Igor Grešovnik: *Programmers' guidelines for Development of Software within COBIK & Laboratory for Multiphase Processes*. Treatise, COBIK, 2012.

[3] Igor Grešovnik: *IoptLib*, electronic document at http://www2.arnes.si/~ljc3m2/igor/ioptlib/.

[4] Igor Grešovnik: *IGLib.NET*, electronic document at http://www2.arnes.si/~ljc3m2/igor/iglib/index.html.