Software Engineering meets Scientific Computing – Group Projects in CSE Education

Michael Bader, Hans-Joachim Bungartz, Ioan Lucian Muntean, Tobias Neckel, and Tobias Weinzierl

Technische Universität München, Institut für Informatik,
Boltzmannstr. 3, 85748 Garching, Germany

Abstract. We present experiences with an annual group project with a strong software engineering touch, recently established in the curriculum of the BGCE honours program, and its impact on the CSE education. The participating students organize themselves as a small software company and work on a topic provided by a specific customer of the project. For example, this year's project aims to develop a flexible geometry handling component to be used in fluid-structure-interaction simulations.

1 Introduction

The Bavarian Graduate School of Computational Engineering (BGCE) is an additional educational program for elite students enrolled in one of the following three master programs: Computational Science and Engineering (CSE) and Computational Mechanics at TU München, and Computational Engineering at University Erlangen-Nürnberg. The BGCE curriculum is based on three pillars: advanced topical courses and summer academies, soft skills seminars, and the so-called *honours project*. We implemented this honours project as a large software development project. Its format was inspired from the University of Stuttgart, where such projects (see [1]) are successfully run as a part of the Software Engineering curriculum [2]. Maybe quite common in computer science programs, this sort of student work is rather new in CSE education.

Our project aims to provide to the CSE students an environment for experiencing and practicing software engineering techniques applied to concrete simulation projects. The student team acts like a small software company, having management staff, scientific and programming experts. The project is treated as a request coming from a real customer, either from a research group or from a company. Therefore, the success of each such project is partially measured by the customer's satisfaction.

Within this frame, we foster the improvement/development of practical skills in software engineering, but also in additional fields such as team work or communication. Thus, we merge in an applied way software engineering and CSE education.

2 The BGCE software development project

The honours project, though running in an academic environment, tries to reflect as much as possible the real-life process of making software. Thus, it provides experience in software engineering techniques in a *learning by doing* manner.

The student team, which consists of 4–6 participants with different background, receives the task to design and implement a well specified piece of software in a limited amount of time (6–8 months). The individual effort is roughly equivalent to one third of a master thesis (10 ECTS credits).

The internal organization of the student team follows one of a small software company with a project manager, scientific and technical experts, well-defined roles and responsibilities. The students themselves have to structure both the work and the effort, so that the project can be completed in time. At the beginning, the team receives an initial requirements document from the customer. Based on that, the team elaborates a concrete design concept, which is followed by a formal contract between the two parties, including functionalities, milestones, releases, deadlines.

Not only writing performant code is important in this kind of project, but also elaborating a proper software design, generating useful documentation, dealing with classical software development techniques, or respecting the terms of a contract. The existence of a customer, who is not directly involved in the software development, but decides in the end upon the success of the project, proves to be a crucial element in achieving the goals of this kind of project. The customer remains independent, unaffected by details, and is purely result-oriented. Thus, the project gets a strong product character.

The theoretical knowledge needed by the students is partially covered by courses in the curriculum of the regular master program and complemented by additional introductory lessons, if required. Moreover, a team of 2-4 researchers accompanies the students throughout their work, providing assistance on scientific or technical issues.

3 The PeGSI project

We ran two such projects in the last years: The first one dealt with the *from-the-scratch* development of a software package (*MolDyn*) for performing and visualizing results of molecular dynamics simulations. The second one (*PeGSI* – Peano Geometry Sophisticated Interface), which finished in April 2007, tackled issues related to the geometry handling in fluid-structure-interaction (FSI) simulations [3]. Here, in contrast to the MolDyn project, the focus was on *reuse* of specific software libraries and tools, such as Open CASCADE, as well as on interoperation with two different existing CFD simulation codes. Furtheron, we focus on the PeGSI project.

Selected project requirements.
In a partitioned approach for FSI simulations, two independent solvers for the flow and the structure computations are used [4]. Each solver has its own internal

mesh. The simulation data has to be communicated to the central interface (a triangulated surface) for access from the other solver, in order to assure a correct interplay for the overall simulation.

The objective of PeGSI is to efficiently bridge the gap between the internal solver meshes (e.g. Cartesian grids of the flow solvers) and the central triangulated surface, in this FSI context. This implicates a number of functional requirements for the PeGSI software component. It has to be able to create mappings from triangulated surfaces to (possibly Cartesian) solver grids in order to answer geometry queries. The other way round, simulation results such as forces from the CFD side or displacements from the structure solver have to be mapped from the internal (Cartesian) grids onto the triangulated surfaces. Finally, the triangulated surface in PeGSI has to be updated whenever the geometry changes.

Furthermore, PeGSI provides a rich set of features meant to efficiently handle the data, such as: *on-the-fly* and *on-demand* generation of the geometry representations (using space-trees); persistent storage of the geometry information; dimension parameterization (same code for 2D and 3D geometries); functional decomposition and data separation (CFD, CSD, PeGSI).

The FSI simulation scenario.
The concrete FSI simulation scenario in which PeGSI was to prove its applicability is the calculation of the *drift ratchet* [5]. Meso-particles are being transported through special pore geometries by a fluid flow that is alternating its direction periodically. It has been observed experimentally that under certain conditions this motion induces a spatial separation of the meso-particles according to their size. In order to gain insight into these physical phenomena, numerical simulations of this FSI scenario are carried out. Due to this special situation, the structure solver simplifies to a rigid body solver for small spherical particles. This was also implemented within the scope of the PeGSI project.

4 Conclusions

In its present form, the BGCE software development project offers to talented CSE students an environment to experience extensive group work in the field of scientific computing. Although the participants have more CSE than computer science background, they enthusiastically embraced and practiced several software engineering key-concepts. Furthermore, they had the opportunity to train soft skills, for example when dealing with conflicts within the team or with the customer. In fact, we experienced that all these aspects turned the honours project into a major attraction for future students to take part in the entire BGCE honours program.

References

1. Bernreuther, M., Bungartz, H.J.: First experiences with group projects in CSE education. Computing in Science & Engineering **8**(4) (July 2006) 16–25

2. Ludewig, J.: Praktische Lehrveranstaltungen im Studiengang Softwaretechnik: Programmierkurs, Software-Praktikum, Studienprojekte, Fachstudie. 2nd edn. Univ. Stuttgart (2001) [in German].
3. Bamakhrama, M., Guillen, C., Mayer, U., Wang, L.: PeGSI – Peano Geometry Sophisticated Interface (2007) http://http://www.bgce.de/curriculum/projects/pegsi.
4. Brenk, M., Bungartz, H.J., Mehl, M., Neckel, T.: Fluid-structure interaction on cartesian grids: Flow simulation and coupling environment. In Bungartz, H.J., Schäfer, M., eds.: Fluid-Structure Interaction. Volume 53. Springer (2006)
5. Kettner, C., P.Reimann, Hänggi, P., Müller, F.: Drift ratchet. Phys. Rev. E. **61**(1) (2000) 312–323