



11th IPB Erasmus Week

Teaching Crossroads

Edited by

Elisabete Silva

Clarisse Pais

Luís S. Pais

Título: Teaching Crossroads: 11th IPB Erasmus Week
Editores: Elisabete Silva, Clarisse Pais, Luís S. Pais
Edição: Instituto Politécnico de Bragança · 2016
5300-253 Bragança · Portugal
Tel. (+351) 273 303 200 · Fax (+351) 273 325 405
<http://www.ipb.pt>
Execução: Serviços de Imagem do Instituto Politécnico de Bragança
Capa: Soraia Maduro
Tiragem: 40 exemplares
ISBN: 978-972-745-210-1
Online version: <http://hdl.handle.net/10198/11262>

Contents

<i>Elisabete Silva, Clarisse Pais, Luís S. Pais</i> Preface	7
HUMAN GEOGRAPHY	11
<i>Pablo M. Orduna Portús</i> The border culture and the heritage management: from the concept to the reality	13
INFORMATION TECHNOLOGY AND FORENSICS.....	27
<i>Michal Podpora</i> Increasing Students' Interests in Computer Vision Lectures by Using Forensic Examples and Proper Teaching Methodologies	29
LANGUAGE AND CULTURE.....	37
<i>Cláudia Martins</i> Is Mirandese a poor second? – Portuguese linguistic heritage Será l mirandés l pariente probezainas? – Fazienda de la lhéngua en Pertual.....	39
INTENSIVE PROGRAMMES / ERASMUS+ MOBILITY OF INDIVIDUALS.....	61
<i>Luís Frólén Ribeiro, João Eduardo Ribeiro, Carlos Casimiro Costa, António Duarte, Carlos Andrade, Arne Svinth, John Madsen, Morten Thomsen, Kent Smidstrup, Carsten John Jacobsen</i> Product Development and Industrial Processing	63
<i>Rui Pedro Lopes</i> An Integrated Approach to Short-term Mobility for Visiting Lecturers....	69

An Integrated Approach to Short-term Mobility for Visiting Lecturers

Rui Pedro Lopes

rlopes@ipb.pt

*Polytechnic Institute of Bragança
Portugal*

Abstract

Student, teaching and non-teaching staff mobility is a fundamental aspect of internationalization in Higher Education in Europe. There have been several programmes that support mobility, in and outside the European context, such as the Erasmus+ programme. However, mobility is not the single expression of internationalization. Cooperation, joint degrees, intensive programmes for language learning, among others, contribute to a more open, solid and high quality education.

This article describes the process followed in the organization of short-term teacher mobility in the scope of the Erasmus programme, with the intention of providing valuable learning experiences to students of the destination HEI. This happened in 2014, and refers to the experience as a visiting lecturer in a master's degree in computer science. The experience was jointly planned and discussed with the degree coordinator, to provide adequate curricular integration.

The reflection about this experience highlights the idea that mobility should not be done isolated, as a single-episode event, without a work plan, and without the possibility of future collaboration. Teacher mobility should be reconsidered

so that it could effectively contribute to create solid international scientific partnerships and the development of joint pedagogical and professional projects.

Keywords: *Erasmus+ programme, computer science, joint cooperation, Distributed Computing in JavaEE.*

Introduction

The European Higher Education Area (EHEA), since the adoption of the Bologna process eighteen years ago, established the foundation for a more comparable and compatible higher education system. Adopting countries have been pursuing this goal by introducing several, and in some cases, substantial, legal and procedural changes that have been changing the way Higher Education Institutions (HEI) work and that have been contributing for a more open and thorough quality assessment system (European Commission, Executive Agency Education, Audiovisual and Culture, 2015).

The concept of internationalization has been evolving since the Bologna process initial adoption. While initially almost totally focused on student mobility, it is now seen as an opportunity for different forms of cooperation, development and quality assessment. It is recognized that internationalization brings several benefits to all the actors, such as students, staff, institutions and countries.

Mobility has been supported by several specific EU-funded programmes, such as Erasmus, Horizon 2020 (European Commission, 2010), Tempus (European Commission, 2014) or Erasmus Mundus, these two currently under the scope of the broader Erasmus+ programme (European Commission, 2015a). In addition, the Erasmus+ also includes the Lifelong Learning Programme, The Youth in Action Programme, Alfa and Edulink, which were previously independent programmes. Erasmus+ integrates these seven programmes in three key actions:

- Learning Mobility of individuals
- Cooperation for innovation and the exchange of good practices
- Support for policy reform.

The implementation of the programme started successfully in 2014, with the distribution of 75% of the total budget to national agencies, with student and staff mobility as the highest priority action. In spite of this, not many countries developed a national strategy for internationalization (European Commission, Executive Agency Education, Audiovisual and Culture, 2015). This is the case of Portugal, which does not have specific policy goals for internationalization.

When compared with the other countries, Portugal still has a long way to go in terms of expressiveness of mobility. In absolute numbers, Spain is leading with the highest number of students sent and received, followed by France, Germany, the United Kingdom and Italy. In relative terms, Luxembourg, Liechtenstein, Latvia, Lithuania and Spain have the highest percentage of outgoing students (European

Commission, 2015b). Portugal has relevance in terms of staff mobility (fourth) and Intensive Language Courses (third).

Nevertheless, most of the actions, including mobility, cooperation and joint degrees, are being implemented by the HEI. In fact, the institutions are encouraged to adopt and implement their own strategy for internationalization, a recommendation that has a significant expression in Portuguese HEI.

Many institutions have been defining internationalization strategies that include several dimensions, such as student and staff mobility, joint degrees, intensive programmes and others. Usually, the institution's strategy mentions, specifically, the need for integration and optimization of the mobility, with the goal of associating it with other kinds of cooperation. However, the implementation of these strategies is seldom performed in an integrated way, focusing on a single aspect and disregarding the others.

Every year, each institution, faculty or school opens a period for application by students, professors and non-teaching staff. Since financial support is limited, a selection process takes place. Students are drafted according to the number of completed ECTS in the degree, excluding others that were already involved in mobility programmes. Professors and staff are drafted according to the number of times that were involved in mobility programmes, disregarding the existence of previous contacts and projects with the receiving institution.

These criteria rarely follows the strategy defined by the institution, allowing mobility to happen in a single, disconnected and inconsequential episode. It can even happen supporting a mobility action of professors and staff with no idea of destination and with no previous contact or activity plan. There should be a careful and thorough reflection on the draft criteria so that the best applications can be sorted out and selected for funding, thus contributing to improve the impact of the internationalization process of HEI.

One of the main impact of professor's mobility is the possibility to bring to students the knowledge and skills that professors have in specific scientific or technological areas. There are specific work and competences, recognized in professors in European HEI, which can contribute to disseminate knowledge and contribute to build strong partnership and cooperation.

This was the main motivation behind the experience described in the next section. This happened in 2014, and refers to the short-term mobility as a visiting lecturer in a master's degree in computer science. The experience was planned, discussed with the destination degree coordinator, and designed to provide valuable knowledge taking into account the specific learning outcomes of the curricular unit. The specificity of the lecturer's technical knowledge created the mobility opportunity to reinforce the students' knowledge in that particular subject. In addition, meetings with the staff and professors were also planned, to find common research interests and the possibility of further cooperation.

This article reflects the process followed in a short-term teacher mobility in the scope of the Erasmus programme with the intention of providing valuable experience to students of the destination HEI, in integration with the curricular unit learning outcomes and benefiting from the specific knowledge of the visiting lecturer.

Lectures

The mobility experience occurred in 2014, starting with an initial approach to the responsible of the destination institution. This contact was performed by email, and allowed to narrow the focus of the action based on the specific knowledge of the visiting professor. This led to the contact with the degree coordinator, which suggested that the lecture could be performed in the course in “Distributed Computing Techniques” of the master’s programme in Computing and Automation Engineering, a follow-up of software engineering and object-oriented programming.

It was agreed that lectures about Distributed Computing in JavaEE would be very appropriate for the students, with the support of a Software Versioning System, such as Git.

Distributed Computing in JavaEE

The Java phenomenon defines more than a programming language. In addition to the object-oriented programming language, Java defines three different architectures:

Java Micro Edition – suitable for embedded, mobile and Internet of Things devices, with limited capacity in terms of memory and processing power;

Java Standard Edition – used in desktop computers, provides rich user interface, portability and connectivity for application development;

Java Enterprise Edition – used to develop enterprise wide applications, providing web-based user interface, communication, application life-cycle management, and others.

The latter defines a distributed computing architecture, providing for packaging of distributable components for deployment, a collection of standardized components, containers, and services for creating and deploying distributed applications within a well-defined distributed computing architecture (Yener, Theedom, & Rahman, 2015).

It is based on the concept of multi-tiered architecture, providing specific containers to deploy different aspects of enterprise applications. This allows the separation of concerns in terms of reading and writing data (data access tier), programme logic and processing (business rules tier), user interface (both web-based and desktop based).

Several technologies are available to develop the components of an enterprise application. In a total of six hours, spread through two days, students learned the importance and the architecture of the Java-Server Faces, the Enterprise Java Beans

for business logic implementation, and the Object-relational Mapping between entity objects and relational databases.

Although they already had had some contact with web-based technologies, such as the Hyper-Text Markup Language (HTML), Javascript and the PHP server-side scripting language, the structuring and modularity of the JavaEE platform was something that the students had not yet addressed. Moreover, the possibility of providing different access protocols, without adding extra code was something new, which made them realize the effectiveness of using this technology for building large scale applications. They quickly got comfortable with building scalable web services, independent of the graphical user interface, with less effort than the required by other technologies.

Overall, the experience was relevant and significant for students, although requiring better code organization and management. This led us to study version control systems with special emphasis on the distributed version control system with Git.

Distributed Version Control System with Git

Git is a free and open source distributed version control system. Version control is a system that records changes to a file or set of files over time, so that it is possible to recall specific versions later (Chacon & Straub, 2014). Version control is particularly important for programming and software projects management, which supports team management, developer collaboration and release management.

There are two types of version control systems models. The centralized model requires a server, hosted in a centralized computer, which contains all the versioned files, and clients, running on the software development team computers that check-out files from the centralized server. This setup has several advantages, allowing administrators and developers to know what everyone is doing and providing easy administration of the software repository. However, there are downsides. The single point of failure could compromise the work of everyone, possibly delaying for hours the collaboration between developers and making it possible to lose information in case of disk failure. Examples of centralized version control systems are CVS (Vesperman, 2006) and Subversion (Pilato, Collins-Sussman, & Fitzpatrick, 2008).

The distributed model does not rely on a single, centralized, server. Clients do not checkout working copies from a server. Instead, each client keeps its own replica of the whole repository. If any client malfunctions, the others do not suffer and can continue working and collaborating. Moreover, each developer can keep a private and individual software version control system in his own computer, making it easier to keep a record of the development history and versions. Examples of distributed version control systems are Git (Chacon & Straub, 2014) and Mercurial (O'Sullivan, 2009).

According to the advantages of distributed version control systems, it was decided to explore, in more detail, how Git can be used and how it can contribute to better programming and software development. Even if they will not use it professionally

in the future, students benefit from this knowledge because of the prevalence of programming languages in several curricular units; version control systems enables them to better organize the code they produce, and to do it faster.

Students learned how to create a distributed repository in their computer, how to commit the code to the repository and keep all the versions as they progress in the development of an application. They also learned how to roll back to a previous version and how to build test areas without interfering with the stable development.

Final Considerations

The EHEA defines and relies on a comparable and comprehensive higher education system, transversal to several countries. This paves the way for unrestricted mobility of students, teaching and non-teaching staff with several benefits to all the actors. Students experience new cultures, new ways to learn, new opportunities that contribute to a better citizenship. Non-teaching staff learn new methods of doing, new organization structures and new processes. Teachers have the opportunity to contribute to stimulate students, sharing his specific knowledge and, at the same time, understand how students from different countries build specific knowledge and how they react to a different learning approach.

Mobility contributes to the overall internationalization experience, fostering cooperation and experience exchange. This, however, should not be done isolated, as a single-episode event with no other consequences. The process of granting support to teacher mobility by HEI should take into consideration the existence of a work plan, the relevance of previous contacts and the possibility of future collaboration. The simple draft based on the number of mobility each teacher does not contribute to strengthen the internationalization success of the institutions and to contribute to effective cooperation.

References

- Chacon, S. & Straub, B. (2014). *Pro Git* (2nd ed. 2014 edition). Apress.
- European Commission. (2015a). *Erasmus+ Programme Annual Report 2014*. Retrieved (03-05-2016) from http://ec.europa.eu/education/library/statistics/erasmus-plus-annual-report_en.pdf
- European Commission (2015b). *Erasmus: Facts, Figures & Trends; The European Union support for student and staff exchanges and university cooperation in 2013-2014*. Retrieved (03-05-2016) from http://ec.europa.eu/education/library/statistics/erasmus-plus-facts-figures_en.pdf
- European Commission (2010). *Estratégia Europa 2020*. Retrieved (03-05-2016) from http://ec.europa.eu/europe2020/index_en.htm
- European Commission, Executive Agency Education, Audiovisual and Culture & Eurydice (2015). *The European higher education area in 2015: Bologna process*

implementation report. Luxembourg: Publications Office of the European Union. Retrieved (03-05-2016) from http://eacea.ec.europa.eu/education/eurydice/documents/thematic_reports/182EN.pdf

European Commission (2014). *The main achievements of the Tempus Programme in Central Asia 1994-2013: from Tempus to Erasmus+*. Brussels: European Commission. Retrieved (03-05-2016) from <http://bit.ly/1X9UNny>

O'Sullivan, B. (2009). *Mercurial: The Definitive Guide* (1 edition). Sebastopol: O'Reilly Media.

Pilato, C. M., Collins-Sussman, B., & Fitzpatrick, B. W. (2008). *Version Control with Subversion* (2 edition). O'Reilly Media.

Vesperman, J. (2006). *Essential CVS* (2nd edition). Beijing ; Cambridge: O'Reilly Media.

Yener, M., Theedom, A. & Rahman, R. (2015). *Professional Java EE Design Patterns* (1st edition). Indianapolis: Wrox.