Chapter 7
An Effective Way to Teach Language Processing Courses

Maria João Varanda Pereira
Instituto Politécnico de Bragança, Portugal

Nuno Oliveira
Universidade do Minho, Portugal

Daniela da Cruz
Universidade do Minho, Portugal

Pedro Rangel Henriques
Universidade do Minho, Portugal

ABSTRACT

All of us that teach Language Processing topics are aware that a great part of the students face big difficulties and a lack of motivation inherent to the concept abstraction level and to the technical capacities required to implement efficient processors. In order to overcome this problem, a starting point is to identify the main concepts involved in Language Processing subject and to consider that a person learns when he/she is involved in a process. The authors argue that motivation is a crucial factor to engage students in the course work, and it is highly dependent on the languages used to work on during the course. Therefore, they discuss the characteristics that a language should have to be a motivating case study. The authors think that LP teachers should be very careful in their choices and be astute in the way they explore the underlying grammars along the course evolution.

1. INTRODUCTION

Learning was, is and will be difficult. The student has to interpret and understand the information he got, and then he has to assimilate the new information merging it with his previous knowledge to generate new knowledge.

DOI: 10.4018/978-1-4666-7304-5.ch007

However teaching is becoming more and more difficult as new student generations are no more prepared to absorb information during traditional classes.

Both statements are true in general, but they are particularly significant in domains that require a high capability for abstraction and for method-

ological analysis and synthesis. This is the case of Computer Science (CS), in general, and of Language Processing (LP) in particular.

As we will show in the sequel, many other authors, researching and teaching in LP domain, have recognized the difficulties faced by both students and teachers. To overcome these difficulties, which frequently lead to the failure and dissatisfaction of all the participants in the learning activity, and keeping in mind that higher education should focus on improving students' problem solving and communication skills, three main approaches can be identified:

- Exploring different teaching methodologies;
- Choosing motivating and adequate languages to illustrate concepts and to create project proposals;
- Resorting to specific tools tailored to support the development of grammars and language processors in classroom context.

As previously introduced in (Varanda Pereira, Oliveira, Cruz, & Henriques, 2013), our focus is the second approach. Considering that *a person just learns when he is involved in a process*, we argue that motivation is a crucial factor to engage students in the course work allowing them to achieve the required knowledge acquisition. In this chapter, we show that motivation is highly dependent on the languages used to work on during the course. We will discuss the characteristics that a language should have to be a motivating case study. LP teachers should choose carefully the sample languages used to explore the underlying grammars along the knowledge transfer process.

Li (2006) states that most topics in a compiler course are quite theoretical and the algorithms covered are more complex than those in other courses. Usually the course content contributes to the lack of student's motivation, giving rise to the student's fail and to the teacher frustration. To improve teaching and learning, there are some effective approaches such as *concept mapping*, *problem solving*, *problem-based learning*, *case studies*, *workshop tutorials* and *eLearning*. In particular Problem-based Learning enables students to establish a relation between abstract knowledge and real problems in their learning. It can increase their interest in the course, their motivation to learn science, make them more active in learning, and improve their problem solving skills and lifelong learning skills. Problem-based Learning is a student-centered teaching approach; however, it was shown (Li, 2006) that the approach gets better results when enrolling students that are not at the first year.

Project-based Learning is another relevant approach to teach compilers. Although similar, Project-based and Problem-based Learning are distinct approaches. In Problem based, the teacher prepares and proposes specific problems (usually focused in a specific course topic, and smaller in size and complexity than a project) and the students work on each one, over a given period of time, to find solutions to the problems; after that, the teacher provides feedback to the students. In Project-based Learning the students, more than solve a specific problem, have to control completely the project; usually the project covers more than one topic and run over a larger period of time.

Islam and Khan (2005) also agree with the complexity of the compiler course and consequently with the students difficulties in this subject. They propose an approach based on templates. Since the automatic construction of compilers is a systematic process, the main idea is to give students templates to produce compilers. The students just have to fill the parts necessary to implement the syntax and the semantics of the language.

Some other authors deal with the problem choosing carefully the language they use for the illustration of concepts or for exercises/projects, as we describe below.

Henry has published a paper (Henry, 2005) about the use of Domain Specific Languages for teaching compilers. He says that building a compiler for a domain specific language can engage