PDS16inEcplise – An Eclipse Plug-in for the PDS16 Assembly Language

T. Oliveira, A. Ramanlal, T. Dias, P. Sampaio
ADEETC, ISEL – Instituto Superior de Engenharia de Lisboa, Instituto Politécnico de Lisboa, Portugal
40653@alunos.isel.ipl.pt 39204@alunos.isel.ipl.pt tdias@cc.isel.ipl.pt psampaio@cc.isel.ipl.pt

Abstract — **This paper presents PDS16inEclipse, a tool to optimize the development of programs for the PDS16 processor. The tool consists of an Eclipse plug-in of a custom text editor that was generated using the Xtext framework, by specifying the Xtext grammar of the PDS16 assembly language and generating the corresponding Eclipse Modeling Framework (EMF) model. Such editor includes several features to facilitate the software development, such as content assist with auto-complete and annotations, real-time syntax and semantic error checking, syntax highlighting and an outline view. To further increase the productivity of the programmers, the tool also provides integration with the existing PDS16 assembler and supports the corresponding error marking feature.**

Keywords: PDS16 Assembly; DSL; Xtext; Eclipse; Plug-in.

I. INTRODUCTION

The *Processador Didático Simples a 16 bits* (PDS16) is a 16-bits RISC-alike processor [1] that was developed at the Instituto Superior de Engenharia de Lisboa (ISEL) in 2008 with two main goals: 1) to present a very simple and educational computer architecture to the students of the second semester of the electrical and computer engineering undergraduate courses, so that they are able to more easily and better understand the fundamental concepts of computer organization and computer architecture, with a special emphasis in assembly programming; and 2) to teach such concepts using a practical approach, by providing a hands-on hardware platform for the students to implement and experiment their programs.

The programs that the students are asked to develop must be written in the PDS16 assembly language [1], for which generic text editors are currently used (e.g. Notepad). Such programs are subsequently translated to the PDS16 machine language either manually (in the beginning of the semester) or by using a custom symbolic command-line assembler that has been developed for the Microsoft Windows operating system [1]. Program verification and validation are performed both in the students' PCs using an emulator and directly on the hardware using the PDS16 development board [1].

Despite its educational goals, this development cycle has proven to be quite limitative when it comes to the final stage of the semester, in which the students are required to develop more complex programs. The reasons are twofold. On the one hand, the use of a standard text editor significantly slows down the program implementation, since the students cannot make use of some basic tools to assist and speedup their task, e.g. syntax highlighting, code completion or source code navigation. On the other hand, the lack of integration between the text editor and the existing PDS16 assembler also contributes to slowing down the program development, since the existence of errors in the build process implies analyzing the log file generated by the assembler to understand the cause of such errors, going back to text editor to implement the necessary modifications to the program description and invoking the assembler again, which involves a change of context for each of these tasks.

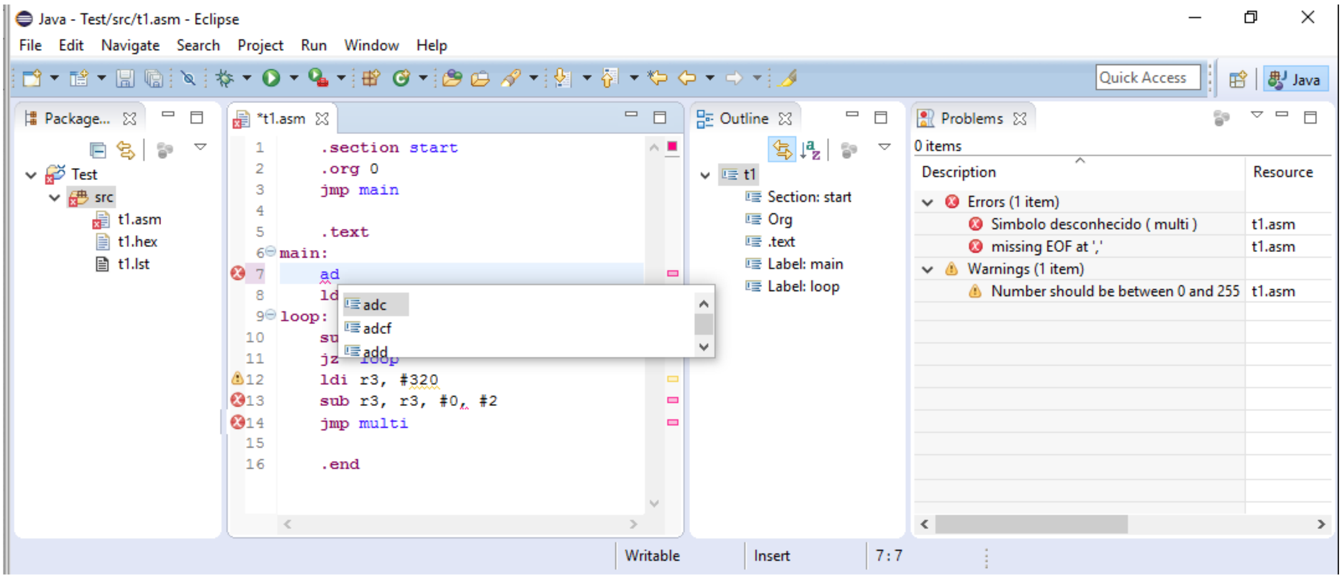
This paper presents a tool that we have developed to overcome these problems [2], and thus optimize the development of programs written in the PDS16 assembly language. This tool was developed using the Xtext framework and consists of a plug-in for the Eclipse platform that provides a custom text editor with the following features: *i*) real-time syntax and semantic error checking; *ii*) content assist with auto-complete and annotations; *iii*) syntax highlighting; *iv*) outline view; and *v*) integration with the existing PDS16 assembler.

The remainder of this paper is organized as follows. In Section II, the Eclipse platform is briefly reviewed. Moreover, a short overview of the Xtext framework is also provided. Then, we present our tool in Section III.

II. DOMAIN-SPECIFIC LANGUAGES AND THE ECLIPSE IDE

Eclipse [3] represents a widely used open development platform based on the Java programming language that is available for multiple platforms and that supports the development of software using various programming languages in an Integrated Development Environment (IDE). Nevertheless, it provides also a flexible and powerful framework for building custom programming language editors with similar functionality to commercial professional IDEs. This is a result of its highly modular architecture that is composed of multiple plug-ins and of the Platform Runtime, which is the core program responsible for the control and for providing services for the plug-ins. The plug-ins are external software tools that are fully integrated in the platform and that provide additional functionality. They are developed using the Plug-in Development Environment (PDE) (also a part of Eclipse) and not only can use the services provided by other plug-ins but also extend their functionality, as well as provide services for other plug-ins. Hence, the plug-ins are used to extend Eclipse's functionality.

Using this model, several tools and software development frameworks have been made available in the Eclipse ecosystem. For example, the Eclipse Modeling Framework (EMF) [4] consists of a modeling and code generation facility for building tools and other applications based on a



```
    .section start
    .org 0
    jmp main

    .text
main:
    ad
    ld
loop:
    su
    jz  loop
    ldi r3, #320
    sub r3, r3, #0, #2
    jmp multi

    .end
```

0 items

Description	Resource
Errors (1 item)	
Simbolo desconhecido (multi)	t1.asm
missing EOF at ','	t1.asm
Warnings (1 item)	
Number should be between 0 and 255	t1.asm

Fig. 1- PDS16inEclipse plug-in look and functionality.

structured data model, which is the basis for several Eclipse projects. One of such projects is Xtext [5], which is an open-source project for developing programming languages and Domain-Specific Languages (DSLs).

Xtext is a high-level framework that covers all the aspects of the development of a complete language infrastructure, from the definition of the DSL grammar, on the parser and the interpreter to code generators and compilers. To achieve such goal, several internal and external tools are employed. For example, the grammar of the DSL is written using the Xtext meta-language, which is a textual language, while the code generation is done with a Java like language named Xtend [5]. Conversely, the ANother Tool For Language Recognition (ANTLR) [6] is used for creating a parser for text processing and Java is employed for performing the constraints validation. Also, Xtext uses the core libraries of Eclipse (i.e. org.eclipse.ui, org.eclipse.jface and org.eclipse.core) for the interface development, which enables the integration of icons and the complete editor management by writing only a minimal amount of code. As a result, with Xtext not only DSLs can be created in a formal and simple way but also fully featured IDEs based on the Eclipse platform can be easily generated.

III. IMPLEMENTATION

The implementation of the PDS16inEclipse tool comprehended several distinct steps. First, we created the grammar for the PDS16 assembly language [1] by using an EBNF-like syntax [7]. With this grammar, we then used the Xtext generator to obtain an ANTLR parser for the PDS16 assembly language with support for the desired syntax checking, syntax highlighting and code completion functionalities. Furthermore, a meta-model of the language that included its Abstract Syntax Tree (AST) was also generated in the form of an EMF model. The Java classes and APIs that were automatically generated with such meta-model were subsequently customized, in order to obtain the semantics checking functionality and the Eclipse Outline view. Finally, the integration with the existing PDS16 assembler was achieved by developing a custom implementation for the code generator stub that is also automatically generated by Xtext. With this approach, every time a PDS16 assembly file is saved in the editor, the Builder automatically calls this generator, which in turn calls the assembler and analyzes its output after its completion. If such output contains error messages, they are presented in the Problems view and the corresponding lines of the source file are also marked in the editor as Eclipse Environment errors.

Fig. 1 shows the look and the functionality of the developed PDS16inEclipse tool, which has been made available online on the project page: http://tiagojvo.github.io/PDS16inEclipse.

REFERENCES

[1] J. Paraíso, "Arquitetura de Computadores," Lecture Notes, ISEL, Lisbon, 2011.

[2] A. Ramanlal and T. Oliveira, "PDS16inEcplise," BSc. Thesis, ADEETC, ISEL, Lisbon, 2016.

[3] D. Carlson, *Eclipse Distilled*, Addison-Wesley Professional, 2005.

[4] D. Steinberg, F. Budinsky, E. Merks and M. Paternostro, *EMF: Eclipse Modeling Framework 2.0*, 2nd Ed., Addison-Wesley Professional, 2009.

[5] L. Bettini, *Implementing Domain-Specific Languages with Xtext and Xtend*, Packt Publishing, 2013.

[6] T. Parr, *The Definitive ANTLR 4 Reference*, The Pragmatic Bookshelf, 2013.

[7] J. W. Backus, F. L. Bauer, J. Green, C. Katz, J. McCarthy, A. J. Perlis, H. Rutishauser, K. Samelson, B. Vauquois, J. H. Wegstein, A. van Wijngaarden, M. Woodger. Report on the algorithmic language ALGOL 60. *Communications of the ACM Magazine*, 3(5): 299-314, May 1960.