

Título: Programação e Simulação de Controladores Lógicos Programáveis - Uma Metodologia, Um Sistema
Title: *Programming and Simulation of Programmable Logic Controllers - a Methodology and a System.*

Autor/Author: José Luís Costa Pinto de Azevedo

Orientador/Advisor: João Pedro Estima de Oliveira

Data Apresentação/Acceptance Date: 11/98

Palavras Chave: Controladores lógicos programáveis, *grafcet*; diagramas *ladder*, simulação, linguagens de programação, programação orientada para objectos

Key Words: *Programmable logic controllers, grafcet, ladder diagrams, simulation, programming languages, object-oriented programming*

Doutoramento/Ph.D.

Resumo

Os controladores lógicos programáveis detêm um papel de grande relevo no meio industrial, sendo aí utilizados em variadíssimas situações. Tomando como base um conceito comum, a grande quantidade de fabricantes, que hoje em dia existe, utiliza consolas de programação e/ou pacotes de *software* proprietários, implementando, por vezes, linguagens aceites de forma idiossincrática.

Esta tese apresenta uma metodologia e um sistema, cujo objectivo é a programação universal de controladores lógicos programáveis. O aspecto essencial da metodologia que se propõe consiste na subdivisão do sistema de programação em dois blocos fundamentais: um *front-end* genérico, não comprometido com qualquer máquina real, com capacidade de edição, geração de código e simulação; e um segundo bloco constituído por um conjunto de tradutores, em número igual ao de diferentes controladores que se pretende programar, que incorporam as características específicas de cada um deles. A comunicação entre estes dois blocos fundamentais é assegurada por meio de uma representação literal obtida pela tradução do programa do utilizador numa linguagem que designámos por Linguagem Literal Intermédia (LLI). Como linguagem de programação-base do sistema é adoptado o *Grafcet*. São implementadas as extensões que facilitam a decomposição hierárquica de uma aplicação (macro-etapa e macro-acção), que tornam esta linguagem uma ferramenta poderosa no controlo de sistemas de eventos discretos, mesmo que de grande dimensão.

Faz-se um estudo da linguagem *Grafcet*, partindo dos conceitos fundamentais das redes de Petri. São discutidos, com especial ênfase, os aspectos não consensuais deste formalismo, no que respeita, nomeadamente, aos algoritmos de interpretação e à utilização de macro-acções. A metodologia e as opções fundamentais que conduziram à implementação do sistema são depois abordadas. Destaca-se a apresentação da arquitectura-base do sistema, os requisitos da LLI e os algoritmos de conversão a ela associados, bem como a arquitectura genérica dos módulos tradutores de código. Faz-se, finalmente, uma descrição dos tópicos essenciais relativos

à implementação do bloco principal do sistema, nas suas vertentes de edição, geração de código e simulação.

Abstract

Programmable Logic Controllers (PLC) play an important role in the industrial environment, being used in many different situations. Taking a common concept, the many existing PLC manufacturers use proprietary programming tools and/or software packages, implementing, some times, usual languages in a idiosyncratic way.

*This thesis presents a methodology and a system, whose purpose is the universal programming of programmable logic controllers. The main concept of the proposed methodology is based on the division of the system into two fundamental blocks: a generic front-end, that is, not committed to any real PLC, allowing program edition, code generation and simulation; a second block composed of a set of code generators (translators), one for each target PLC to be programmed, which assume their specific characteristics. The communication between these two main blocks is accomplished by means of a literal representation that results from the translation of the user program into a language that we called Intermediate Literal Language (ILL). The system takes the graphical language *Grafcet* as the main programming language. The two extensions that allow hierarchical specification of an application (macro-step and macro-action) are implemented. This way, being a formalism well suited to the description of sequential tasks, *Grafcet* becomes a powerful tool in the control of event-driven systems.*

*Taking the fundamental concepts of Petri nets, a study of the *Grafcet* language is made. Non consensual behavioural aspects of such formalism are discussed, regarding, namely, evolution algorithms and the use of the macro-action concept. The methodology and the key options that led to the implementation of the system are then presented. The main topics discussed are the system fundamental architecture, the ILL requisites and the related translation algorithms, as well as the generic architecture of the target PLC code generators. Finally, a description of the main implementation topics of the system front-end is presented, concerning its three main activities: edition, code generation and simulation.*