

Ferramenta Educacional para o Ensino do GRAFCET

Por
Elton Ribeiro Matias

Orientador: José Paulo Barroso de Moura Oliveira

Co-orientador: José Boaventura da Cunha

Dissertação submetida à
UNIVERSIDADE DE TRÁS-OS-MONTES E ALTO DOURO
para obtenção do grau de
MESTRE
em Engenharia Electrotécnica e de Computadores, de acordo com o disposto no
DR – I série–A, Decreto-Lei n.º 74/2006 de 24 de Março e no
Regulamento de Estudos Pós-Graduados da UTAD
DR, 2.ª série – Deliberação n.º 2391/2007

Ferramenta Educacional para o Ensino do GRAFCET

Por
Elton Ribeiro Matias

Orientador: José Paulo Barroso de Moura Oliveira

Co-orientador: José Boaventura da Cunha

Dissertação submetida à
UNIVERSIDADE DE TRÁS-OS-MONTES E ALTO DOURO
para obtenção do grau de
MESTRE
em Engenharia Electrotécnica e de Computadores, de acordo com o disposto no
DR – I série–A, Decreto-Lei n.º 74/2006 de 24 de Março e no
Regulamento de Estudos Pós-Graduados da UTAD
DR, 2.ª série – Deliberação n.º 2391/2007

Orientação Científica :

José Paulo Barroso de Moura Oliveira

Professor Auxiliar com Agregação do
Departamento de Engenharias, Escola de Ciências e Tecnologia
Universidade de Trás-os-Montes e Alto Douro

José Boaventura da Cunha

Professor Auxiliar do
Departamento de Engenharias, Escola de Ciências e Tecnologia
Universidade de Trás-os-Montes e Alto Douro

À Graziella
Aos meus pais

Elton Matias

A quem dedico,
este trabalho.

UNIVERSIDADE DE TRÁS-OS-MONTES E ALTO DOURO
Mestrado em Engenharia Electrotécnica e de Computadores

Os membros do Júri recomendam à Universidade de Trás-os-Montes e Alto Douro a aceitação da dissertação intitulada “**Ferramenta Educacional para o Ensino do GRAFCET**” realizada por **Elton Ribeiro Matias** para satisfação parcial dos requisitos do grau de **Mestre**.

Julho 2009

Presidente: **Salviano Soares Filipe Pinto Soares,**
Direcção do Mestrado em Engenharia Electrotécnica e de
Computadores do Departamento de Engenharias da Universidade
de Trás-os-Montes e Alto Douro

Vogais do Júri: **José Carlos Meireles Monteiro Metrôlho,**
Professor-adjunto do Instituto Politécnico de Castelo Branco

José Paulo Barroso de Moura Oliveira,
Professor Auxiliar com Agregação do Departamento de
Engenharias, Escola de Ciências e Tecnologia da Universidade de
Trás-os-Montes e Alto Douro

José Boaventura da Cunha,
Professor Auxiliar do Departamento de Engenharias, Escola de
Ciências e Tecnologia da Universidade de Trás-os-Montes e Alto
Douro

Ferramenta Educacional para o Ensino do GRAFCET

Elton Ribeiro Matias

Submetido à Universidade de Trás-os-Montes e Alto Douro
para o preenchimento dos requisitos parciais para obtenção do grau de
Mestre em Engenharia Electrotécnica e de Computadores

Resumo — A programação de Controladores Lógicos (Programmable Logic Controller, PLC), é um tópico de muito interesse quer do ponto de vista académico, quer pelas inúmeras aplicações de âmbito industrial. Estes dispositivos permitem a implementação de controladores lógicos no contexto do controlo de sistemas de eventos discretos pela utilização da linguagem GRAFCET. Uma grande parte dos fabricantes de PLC's inclui o GRAFCET como linguagem de programação. O ensino do GRAFCET é fulcral no âmbito de unidades curriculares de Automação Industrial. As tecnologias multimédia associadas ao método de ensino tradicional constituem uma mais valia no processo de ensino/aprendizagem. O desenvolvimento de ferramentas de software que permitam complementar o ensino deste tópico é muito útil.

Neste trabalho foi desenvolvida uma ferramenta educacional, e-GRAFCET, para o apoio e auto-aprendizagem do Grafcet que permite complementar o seu ensino. O trabalho realizado aborda, numa fase inicial, o GRAFCET como ferramenta de modelação de sistemas de eventos discretos. São apresentados alguns editores de *software* de desenho do GRAFCET e *software* de automação que permitem elaborar programas em Grafcet. São apresentadas ferramentas e sítios existentes na Internet similares ao e-GRAFCET e demonstrado o interesse e importância no desenvolvimento de ferramentas deste género.

É apresentada uma descrição da estrutura e desenvolvimento do e-GRAFCET, assim como a justificativa das opções tomadas para a sua construção, analisando todo o seu conteúdo. Para a validação desta ferramenta, foi proposto a um conjunto de alunos, a realização de um inquérito. Este permitiu testar a ferramenta desenvolvida e verificar a sua utilidade e contributo no ensino e na aprendizagem dos alunos, bem como recolher informação para melhorá-la. A análise dos resultados obtidos neste inquérito mostra de uma forma inequívoca a valia do e-GRAFCET como ferramenta educativa. Esta ferramenta vai permitir um ensino mais dinâmico e motivador, sendo um bom utilitário à melhoria da qualidade do processo ensino/aprendizagem.

Palavras Chave: GRAFCET, Automação Industrial, Controladores Lógicos Programáveis, Sistemas de Eventos Discretos, Aprendizagem à Distância.

Educational Tool for Teaching GRAFCET

Elton Ribeiro Matias

Submitted to the University of Trás-os-Montes and Alto Douro
in partial fulfillment of the requirements for the degree of
Master of Science in Electrical and Computers Engineering

Abstract — Programming logic controllers (Programmable Logic Controllers, PLC), it is a very interesting topic, either from the academic perspective or by its innumerable industrial applications. These devices allow the implementation of logic controllers in the context of discrete event control systems by using the GRAFCET language. A large amount of PLC manufacturers incorporate the GRAFCET as a programming language. Teaching GRAFCET is crucial within Industrial Automation courses. Associating multimedia technologies with classical teaching methodologies constitute a great improvement in the process of teaching/learning. The development of software tools which allows complementing the teaching of this topic is very useful.

In this work an educational tool was developed, e-GRAFCET, for supporting the self-learning process of GRAFCET, and allows complementing its teaching. The developed work addresses, in an initial section, the GRAFCET as a modeling tool for discrete event systems. Some software editors for designing GRAFCET and programming with GRAFCET are reviewed. Software tools and sites available in the Internet, which are similar to e-GRAFCET are also reviewed, and the interest of development of tools like these stated.

The description of the structure and development process of the e-GRAFCET is presented, justifying the options used by analyzing its contents. To validate this tool, an inquire was performed within students. This allowed to test the developed tool and to verify its usefulness and contribution in the learning process of students, as well as to gather information in order to improve it. The inquire results analysis shows without doubt the usefulness of e-GRAFCET as a teaching tool. This tool will allow a teaching process more dynamic and challenger, as a good tool to improve the quality of teaching/learning.

Key Words: GRAFCET, Industrial Automation, Programmable Logic Controllers, Discrete Event Systems, e-Learning.

Agradecimentos

Ao longo do meu trabalho, alguns foram os que contribuíram com a sua ajuda e motivação, sem as quais o presente trabalho não teria sido possível.

A realização desta Dissertação de Mestrado contou em primeiro lugar com a ajuda e orientação do Professor Doutor José Paulo Barroso de Moura Oliveira, que com a sua dedicação, disponibilidade, experiência, constante motivação, numerosas sugestões e críticas, foram uma mais valia para que esta dissertação se tornasse uma realidade. Por tudo isto, Professor, bem haja.

Ao Professor Doutor José Boaventura da Cunha por todo o apoio e acompanhamento ao longo da realização deste trabalho.

Aos alunos, pela colaboração na validação da ferramenta. A todos aqueles que de alguma forma contribuíram para a realização deste trabalho, quero deixar aqui os meus sinceros agradecimentos.

À minha família e amigos, todo o apoio que sempre me deram e por ter compreendido os vários momentos em que não pude estar presente.

Finalmente, o meu especial agradecimento à minha namorada, que pela sua motivação, ajuda e apoio incondicional nos bons e menos bons momentos por que passámos, tornaram possível a concretização deste trabalho.

A todos, bem hajam !

UTAD, Vila Real
30 de Julho, 2009

Elton Ribeiro Matias

Índice

| | |
|---|----------|
| Resumo | i |
| <i>Abstract</i> | iii |
| Agradecimentos | v |
| Índice de figuras | xiv |
| Glossário | xv |
| 1 Introdução | 1 |
| 1.1 Motivação | 1 |
| 1.2 Objectivos | 3 |
| 1.3 Organização da dissertação | 4 |
| 2 GRAFCET | 5 |
| 2.1 Introdução | 5 |
| 2.2 Perspectiva histórica | 6 |
| 2.3 O GRAFCET | 7 |
| 2.3.1 Etapas | 8 |
| 2.3.2 Transições e receptividades | 9 |
| 2.3.3 Acções | 11 |
| 2.3.4 Regras de evolução | 13 |
| 2.3.5 Estruturas de base | 14 |

| | | |
|-------|--|----|
| 2.3.6 | Estruturação e hierarquia do Grafcet | 18 |
| 2.3.7 | Outras estruturas | 21 |
| 2.4 | Editores de Grafcet | 22 |
| 2.4.1 | SFCEdit | 22 |
| 2.4.2 | GRAFTOR | 22 |
| 2.4.3 | EdiTSAB | 24 |
| 2.4.4 | MACHCET | 24 |
| 2.4.5 | WinGrafcet | 25 |
| 2.4.6 | Grafcet | 25 |
| 2.5 | Software de automação | 26 |
| 2.5.1 | Automation Studio | 27 |
| 2.5.2 | CoDeSys | 28 |
| 2.5.3 | IsaGRAF | 28 |
| 2.5.4 | STRATON Workbench | 30 |
| 2.5.5 | PL7 | 30 |
| 2.6 | Sítios e ferramentas que permitem aprender o GRAFCET | 31 |
| 2.7 | Interesse deste tipo de ferramentas | 34 |
| 2.8 | Notas finais | 35 |

3 e-GRAFCET 37

| | | |
|-------|--|----|
| 3.1 | Introdução | 37 |
| 3.2 | Estrutura do e-GRAFCET | 37 |
| 3.2.1 | E-LEARNING | 37 |
| 3.2.2 | Adobe Flash | 38 |
| 3.2.3 | Adobe Dreamweaver | 39 |
| 3.2.4 | XML | 41 |
| 3.3 | Descrição da ferramenta | 42 |
| 3.3.1 | Página inicial | 42 |
| 3.3.2 | Noções elementares | 42 |
| 3.3.3 | Estruturas elementares | 44 |
| 3.3.4 | Acções | 52 |
| 3.3.5 | Estruturas avançadas | 57 |
| 3.4 | Exemplos | 62 |
| 3.4.1 | Exemplo 1: Ciclo de uma só sequência | 63 |
| 3.4.2 | Exemplo 2: Paralelismo Interpretado | 68 |
| 3.4.3 | Exemplo 3: Paralelismo Estrutural | 71 |
| 3.4.4 | Exemplo 4: Recurso Partilhado | 75 |
| 3.4.5 | Exemplo 5: Prioridades | 79 |
| 3.4.6 | Notas adicionais aos exemplos | 84 |
| 3.5 | Testes sobre grafcet | 84 |

| | | |
|----------|--|------------|
| 3.5.1 | Estrutura do ficheiro XML | 85 |
| 3.5.2 | Funcionamento e interface | 85 |
| 3.6 | Notas finais | 87 |
| 4 | Validação do e-GR AFCET | 89 |
| 4.1 | Introdução | 89 |
| 4.2 | Descrição do inquérito | 89 |
| 4.2.1 | Implementação do inquérito | 90 |
| 4.3 | Análise dos dados | 90 |
| 4.3.1 | Primeira parte | 90 |
| 4.3.2 | Segunda parte | 94 |
| 4.3.3 | Terceira parte | 96 |
| 4.4 | Notas finais | 98 |
| 5 | Conclusão e Trabalho Futuro | 99 |
| 5.1 | Síntese do trabalho | 99 |
| 5.2 | Recomendações de trabalho futuro | 100 |
| 5.3 | Comentários finais | 101 |
| | Referências bibliográficas | 103 |
| A | Inquérito do e-GR AFCET | 109 |

Índice de figuras

| | | |
|------|---|----|
| 2.1 | Linguagens da norma IEC 61131-3. | 7 |
| 2.2 | Representação de etapas. | 8 |
| 2.3 | Representação de transições e receptividades | 9 |
| 2.4 | Transições com receptividade temporizada. | 10 |
| 2.5 | Transição com predicado. | 10 |
| 2.6 | Não simultaneidade da receptividade em flancos. | 11 |
| 2.7 | Representação de uma acção contínua. | 11 |
| 2.8 | Representação de acções condicionadas. | 12 |
| 2.9 | Acção retardada e acção limitada no tempo. | 12 |
| 2.10 | Acção no evento. | 12 |
| 2.11 | Exemplo de uma acção no evento. | 13 |
| 2.12 | Exemplos de acções memorizadas. | 13 |
| 2.13 | Acção no disparo. | 13 |
| 2.14 | Exemplo de sequência e de um ciclo com uma só sequência. | 15 |
| 2.15 | Exemplo de selecção de sequência e reagrupamento de sequências. | 15 |
| 2.16 | Activação de sequências paralelas. | 16 |
| 2.17 | Sincronização de sequências. | 16 |
| 2.18 | Sincronização e activação de sequências paralelas. | 17 |

| | | |
|------|--|----|
| 2.19 | Salto de etapas. | 18 |
| 2.20 | Retoma de sequência. | 18 |
| 2.21 | Exemplos de forçagem. | 19 |
| 2.22 | Encapsulamento. | 20 |
| 2.23 | Representação de uma macro-etapa e macro-expansão. | 20 |
| 2.24 | Etapa fonte e etapa poço. | 21 |
| 2.25 | Transição fonte e transição poço. | 22 |
| 2.26 | Ambiente de trabalho do SFCEdit. | 23 |
| 2.27 | Exemplo de janela do Graftor. | 23 |
| 2.28 | Ilustração retirada do EdiTSAB. | 24 |
| 2.29 | Exemplo retirado do MACHCET. | 25 |
| 2.30 | Exemplo do WinGrafcet. | 26 |
| 2.31 | Ambiente de trabalho do Grafcet. | 26 |
| 2.32 | Ilustração do Automation Studio. | 28 |
| 2.33 | Ambiente de trabalho do CoDeSys. | 29 |
| 2.34 | Ilustração do IsaGRAF. | 30 |
| 2.35 | Ilustração do STRATON. | 31 |
| 2.36 | Ambiente de trabalho do PL7. | 32 |
| 3.1 | Página inicial do e-GRAFCET. | 40 |
| 3.2 | Página de um exemplo prático com ajuda. | 41 |
| 3.3 | Representação dos elementos fundamentais do GRAFCET. | 43 |
| 3.4 | Receptividades associadas às transições. | 44 |
| 3.5 | Evolução do Grafcet - Regra 5. | 44 |
| 3.6 | Divergência OU. | 45 |
| 3.7 | Convergência OU. | 46 |
| 3.8 | Divergência E. | 47 |
| 3.9 | Convergência E. | 47 |
| 3.10 | Escolha de sequência 1. | 48 |
| 3.11 | Escolha de sequência 2. | 49 |
| 3.12 | Retoma de sequência. | 50 |
| 3.13 | Salto de etapas. | 50 |

| | | |
|------|---|----|
| 3.14 | Paralelismo estrutural. | 51 |
| 3.15 | Paralelismo interpretado. | 52 |
| 3.16 | Acção contínua. | 53 |
| 3.17 | Acção condicional. | 54 |
| 3.18 | Acção limitada. | 55 |
| 3.19 | Acção retardada. | 56 |
| 3.20 | Acção memorizada. | 57 |
| 3.21 | Acção impulsional. | 58 |
| 3.22 | Exemplo de activação forçada de uma etapa. | 58 |
| 3.23 | Exemplo de desactivação de uma etapa poço. | 59 |
| 3.24 | Exemplo de transição fonte. | 59 |
| 3.25 | Exemplo de transição poço. | 60 |
| 3.26 | Exemplo de macro-etapa e de macro-expansão. | 60 |
| 3.27 | Exemplo de sub-programa (T1). | 61 |
| 3.28 | Exemplo de receptividade dependente do tempo. | 63 |
| 3.29 | Exemplo 1. | 63 |
| 3.30 | Ciclo de uma só sequência - Exemplo 1.1. | 64 |
| 3.31 | Ciclo de uma só sequência - Exemplo 1.2. | 65 |
| 3.32 | Ciclo de uma só sequência - Exemplo 1.3. | 66 |
| 3.33 | Ciclo de uma só sequência - Exemplo 1.4. | 67 |
| 3.34 | Exemplo 2. | 68 |
| 3.35 | Paralelismo interpretado - Exemplo 2.1. | 69 |
| 3.36 | Paralelismo interpretado - Exemplo 2.2. | 70 |
| 3.37 | Exemplo 3. | 71 |
| 3.38 | Paralelismo estrutural - Exemplo 3.1. | 72 |
| 3.39 | Paralelismo estrutural - Exemplo 3.2 | 73 |
| 3.40 | Paralelismo estrutural - Exemplo 3.3 | 74 |
| 3.41 | Exemplo 4. | 75 |
| 3.42 | Recurso partilhado - Exemplo 4.1. | 76 |
| 3.43 | Recurso partilhado - Exemplo 4.2. | 77 |
| 3.44 | Recurso partilhado - Exemplo 4.3. | 79 |

| | | |
|------|---|----|
| 3.45 | Exemplo 5. | 80 |
| 3.46 | Prioridades - Exemplo 5.1. | 81 |
| 3.47 | Prioridades - Exemplo 5.2. | 82 |
| 3.48 | Botões de controlo da animação. | 84 |
| 3.49 | Testes online: exemplo de uma pergunta. | 86 |
| 3.50 | Testes online: classificação. | 87 |
| | | |
| 4.1 | Representação gráfica da percentagem de respostas à questão 1. . . . | 91 |
| 4.2 | Representação gráfica da percentagem de respostas à questão 2. . . . | 91 |
| 4.3 | Representação gráfica da percentagem de respostas à questão 3. . . . | 92 |
| 4.4 | Representação gráfica da percentagem de respostas à questão 4. . . . | 93 |
| 4.5 | Representação gráfica da percentagem de respostas à questão 5. . . . | 93 |
| 4.6 | Representação gráfica da percentagem de respostas à questão 6. . . . | 94 |
| 4.7 | Representação gráfica da percentagem de respostas à questão 10. . . . | 95 |

Glossário

| Acrónimo | Descrição |
|-----------------|---|
| <i>ADEPA</i> | Agence nationale pour le Développement de la Production Automatisée |
| <i>AFCET</i> | Association Française pour la Cybernétique Économique et Technique |
| <i>CD – ROM</i> | Compact Disc - Read Only Memory |
| <i>FBD</i> | Function Block Diagrams |
| <i>GRAFCET</i> | GRAPhe Fonctionnel de Commande Étape - Transition |
| <i>HTML</i> | HyperText Markup Language |
| <i>IEC</i> | International Electrotechnical Commission |
| <i>IL</i> | Instruction List |
| <i>KB</i> | KiloByte |
| <i>LD</i> | Ladder Diagrams |
| <i>PLC</i> | Programmable Logic Controller |
| <i>SFC</i> | Sequential Function Chart |
| <i>ST</i> | Structured Text |
| <i>UTAD</i> | Universidade de Trás-os-Montes e Alto Douro |
| <i>W3C</i> | World Wide Web Consortium |
| <i>XML</i> | eXtensible Markup Language |



Introdução

1.1 Motivação

A programação de autómatos, vulgos Controladores Lógicos Programáveis (Programmable Logic Controller, PLC) é um tópico de muito interesse quer do ponto de vista académico, quer pelas inúmeras aplicações de âmbito industrial. Um PLC é similar a um computador digital e está desenhado para ser usado em ambientes e processos de automação industrial. Ao contrário dos computadores tradicionais, um PLC tem múltiplas entradas e saídas para dar suporte aos vários sensores e actuadores do sistema a controlar. Estes dispositivos podem controlar diversas máquinas de um sistema. A resistência às altas temperaturas, às perturbações eléctricas e a robustez física aos impactos e vibrações a que está sujeito, são algumas das características deste tipo de dispositivo que o tornam bastante fiável e de utilização segura no meio industrial. O controlo das operações é efectuado pelo programa do autómato que é guardado em memória não volátil (permanente). Existem várias ferramentas que permitem programar autómatos, variando conforme a marca comercial do PLC. Estes dispositivos permitem a implementação de controladores lógicos no contexto do controlo e modelação de sistemas de eventos discretos [1].

Considerando as variáveis binárias como variáveis lógicas, verdadeiro (**1**) e falso (**0**), estes controladores de eventos discretos podem ser descritos por um conjunto de *equações lógicas* que especificam as suas saídas (ou ordens que envia para o sistema a controlar) e valores de *funções Booleanas* das suas entradas (ou informações que recebe do sistema a controlar) [2]. Assim, sistemas deste género são considerados sistemas de tempo real uma vez que alteram as suas saídas em função das entradas que reflectem o estado actual do sistema.

Para tentar normalizar a programação de autómatos foi desenvolvida uma ferramenta de modelação chamada GRAFCET¹ que tem vindo a ganhar popularidade crescente nos últimos 30 anos. O GRAFCET é uma linguagem de especificação gráfica que permite descrever as várias fases de funcionamento de um automatismo, em forma de diagrama. Tem uma boa adaptação e integração nos autómatos programáveis. Além disso, permite visualizar o comportamento de um automatismo sequencial de uma forma simples, clara e de fácil compreensão. Por este motivo, esta ferramenta é tão importante para solucionar o controlo de sistemas de eventos discretos [3] pela modelação de controladores lógicos, que podem ser implementados pelas linguagens de programação [4] nos diferentes tipos de autómatos.

O ensino do GRAFCET é fulcral no âmbito de unidades curriculares de Automação Industrial. Muitas vezes, aprender os conceitos básicos do GRAFCET não é uma tarefa fácil, ainda mais para quem dá os primeiros passos neste vasto "mundo" da automação. A tecnologia multimédia associada ao método de ensino tradicional constitui uma mais valia no processo de ensino/aprendizagem [5].

O desenvolvimento de ferramentas de *software* que permitam complementar o ensino deste tópico são muitos úteis. Este tipo de ferramentas deve ser concebida de forma a permitir uma maior auto-aprendizagem por parte dos alunos. O e-GRAFCET surge para dar um contributo neste domínio de aplicação.

¹GRAphe Fonctionnel de Commande Étape - Transition

1.2 Objectivos

A recente implementação do tratado de Bolonha no curso de *Engenharia Electrotécnica e de Computadores da UTAD*² provocou ajustes significativos nas actuais metodologias de ensino. Isto obrigou a uma definição mais precisa dos objectivos e competências a ser adquiridas pelos alunos, bem como a reformulação de estratégias e métodos de ensino. A actual exigência no ensino impõe que o aluno seja um elemento activo na educação e assimilação do conhecimento [6].

Neste contexto, é necessário o desenvolvimento de novas metodologias e ferramentas de interacção com os alunos, que proporcionem um ensino inovador, agradável e motivador.

O objectivo global desta dissertação de mestrado é desenvolver uma ferramenta de apoio ao ensino da programação de PLC, capaz de transmitir os vários conceitos do GRAFCET por intermédio de simulações.

Os objectivos mais específicos são:

- Descrição do GRAFCET na actual norma em vigor (IEC 61131-3);
- Levantamento do estado da arte e das aplicações de software similares;
- Planificação de um *site*: a ferramenta a desenvolver irá ser executada em linha (*online*) ou em diferido (*offline*) em qualquer *browser*;
- Definição da estrutura a seguir na construção da ferramenta;
- Escolha do software de simulação gráfica que permita a visualização da evolução simultânea quer do modelo do Grafcet, quer da representação gráfica do sistema;
- Desenvolvimento da interface computacional gráfica;
- Escolha de exemplos padrão para as várias estruturas do GRAFCET e sua implementação;

²Universidade de Trás-os-Montes e Alto Douro

- Realização de testes de validação da ferramenta desenvolvida.

1.3 Organização da dissertação

A dissertação está organizada em cinco capítulos. Neste capítulo introdutório expõe-se o problema em questão, as motivações e os objectivos orientadores deste trabalho.

O capítulo 2 aborda os editores de desenho do GRAFCET e *software* de programação. É apresentada uma introdução e descrição do GRAFCET como ferramenta de modelação de sistemas de eventos discretos. De seguida, são descritas as ferramentas e aplicações de software similares. A finalizar o capítulo e de acordo com o tema deste trabalho, é abordado o interesse no desenvolvimento de aplicações deste género.

O capítulo 3 apresenta a descrição do e-GRAFCET, a metodologia utilizada e as opções tomadas quanto ao software usado para desenvolver toda a ferramenta. São explicados os conteúdos todos que constituem a ferramenta e apresentados os casos práticos da aplicação dessas estruturas.

O capítulo 4 descreve a realização dos testes e do processo de validação da ferramenta desenvolvida, por intermédio de um inquérito proposto aos alunos. Este inquérito permitiu avaliar na prática a utilidade e o contributo desta ferramenta para o ensino/aprendizagem, além de que o *feedback* recebido foi relevante para proceder a melhorias e actualizações do e-GRAFCET.

O capítulo 5 é dedicado ao sumário da dissertação, onde são apontadas as conclusões deste trabalho e referem-se algumas directrizes orientadoras de trabalho futuro. Os comentários do autor surgem como nota final do trabalho.



GRAFCET

2.1 Introdução

Neste capítulo é apresentada uma revisão do GRAFCET como ferramenta de modelação de sistemas de eventos discretos.

O desenho e edição do GRAFCET são facilitados por editores de *software*. Estes programas proporcionam um bom contributo na concepção de programas Grafcet simples ou complexos e podem, por exemplo, auxiliar na documentação de um projecto de automação.

A programação de sistemas de controlo sequencial descritos em GRAFCET é realizada através das linguagens mais comuns de programação de PLC [6]. É apresentada uma breve descrição do *software* de automação actual que permitem elaborar programas baseados em Grafcet para a programação de PLC. De seguida é apresentado um levantamento de ferramentas e sites existentes que permitem aprender o GRAFCET.

Neste capítulo é também abordado o interesse e importância no desenvolvimento de ferramentas de *software* que permitam complementar e motivar a auto-aprendizagem do GRAFCET por parte dos alunos.

2.2 Perspectiva Histórica

Na década de 70 do século passado, a avaliação da concepção dos sistemas lógicos não era a melhor [7]. Investigadores trabalhavam para melhorar os métodos teóricos de síntese (Huffman, expressões regulares, etc.) dado ainda terem pouco impacto na indústria [8]. Com a evolução das tecnologias houve a necessidade de entreaajuda entre industriais e investigadores.

Em 1975 foi criada uma comissão com o intuito de normalizar a representação do caderno de encargos de um automatismo lógico, por intermédio do grupo de trabalho intitulado "sistemas lógicos" do AFCET¹ [9]. Esta comissão, formada por investigadores e industriais, estudou os vários modelos com o objectivo de homogeneizar as diferentes abordagens num único instrumento. Após dois anos, em 1977, a comissão apresenta no seu relatório final um novo método de descrição do caderno de encargos dos automatismos lógicos: o GRAFCET [10]. Este método é uma representação gráfica, de fácil leitura e compreensão, que permite descrever todas as funções realizadas pelo automatismo, ou seja, um Diagrama Funcional.

O nome GRAFCET deriva de GRaph (devido ao fundamento gráfico do modelo) e AFCET (associação científica que financiou a pesquisa). Nos anos posteriores verificou-se uma boa aceitação tanto no ensino como na indústria. A ADEPA² responsável por promover a automatização nas pequenas e médias indústrias, retomou os trabalhos da comissão AFCET com o objectivo de dar ao GRAFCET uma forma normalizada. Em 1982, graças ao trabalho da ADEPA, o GRAFCET torna-se uma norma francesa (NF C03-190) [11].

Cinco anos mais tarde, em 1988, o GRAFCET foi adoptado como uma norma internacional pela IEC³ com o nome inglês Sequential Function Chart (SFC) e referência 848. Mais recentemente, com a alteração nas normas IEC passou a ser a norma IEC 60848.

¹Association Française pour la Cybernétique Économique et Technique

²Agence nationale pour le Développement de la Production Automatisée

³International Electrotechnical Commission

A norma IEC 61131-3 [3] introduziu algumas modificações ao padrão IEC 60848, de modo a adaptar esta linguagem às restantes da norma e implementar modernas técnicas de programação para os sistemas de controlo [4, 12]. As cinco linguagens representadas na Figura 2.1 e definidas pela norma são:

- GRAFCET (*Sequential Function Chart* - SFC)
- Lista de Instruções (*Instruction List* - IL)
- Diagrama de contactos (*Ladder Diagrams* - LD)
- Diagrama de blocos de funções (*Function Block Diagrams* - FBD)
- Texto Estruturado (*Structured Text* - ST)

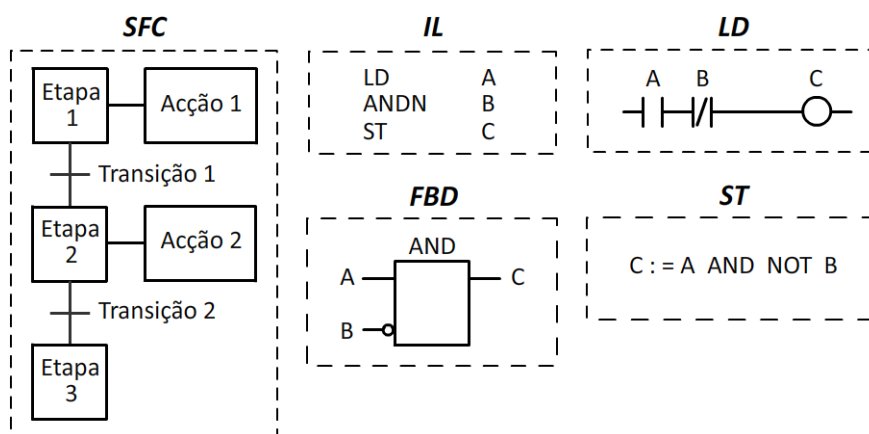


Figura 2.1 – Linguagens da norma IEC 61131-3.

A norma IEC 61131-3 aplica-se aos autómatos programáveis [13] mas não põe em causa nem anula a norma IEC 60848 que continua, por enquanto, a ser o único elemento de referência oficial no que respeita ao GRAFCET.

2.3 O GRAFCET

O GRAFCET é uma linguagem de especificação para a descrição funcional do comportamento determinístico da parte sequencial e da parte de comando de um

sistema de produção automatizado [9]. Nesta secção são apresentados os conceitos fundamentais do GRAFCET.

2.3.1 Etapas

A etapa é utilizada para definir a situação da parte sequencial de um sistema. Exemplos de representação de várias etapas estão ilustrados na Figura 2.2.

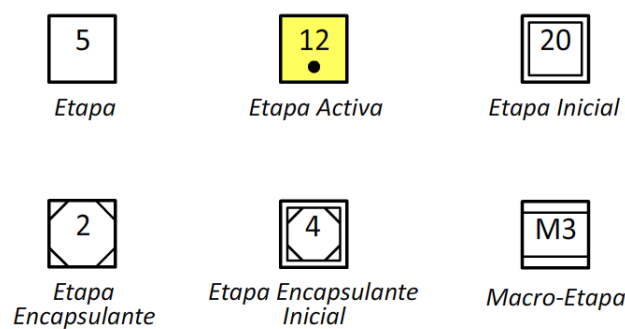


Figura 2.2 – Representação de etapas.

Apresenta-se uma breve descrição de cada uma das etapas representadas na Figura 2.2:

Etapa - Uma etapa pode ser identificada pelo uso de um nome (*label*), ou designação, a qual terá de incluir outros caracteres que não sejam numéricos. O estado da variável lógica Booleana associada a esta etapa é $X5=0$, pois a etapa está desactiva.

Etapa Activa - Uma etapa activa é representada com o uso de uma marca. O estado da variável lógica Booleana associada a esta etapa é $X12=1$.

Etapa Inicial - Esta etapa está activa na situação inicial do Grafcet.

Etapa Encapsulante - Uma etapa encapsulante está associada a um conjunto de etapas ditas encapsuladas.

Etapa Encapsulante Inicial - A activação desta etapa na situação inicial do *Grafcet* implica a activação da ou das etapas encapsuladas marcadas com um

asterisco⁴.

Macro-Etapa - É a representação única de uma parte detalhada do *Grafcet*. A essa representação é denominada *expansão* do Grafcet.

2.3.2 Transições e Receptividades

Uma transição representa a possibilidade de evolução de actividade entre duas ou mais etapas. São apresentadas algumas transições com receptividades mais comuns, Figura 2.3. Entende-se por receptividade, h , uma proposição lógica (booleana) que pode assumir dois valores: verdadeiro ou falso [14]. Na Figura 2.3 são dados alguns exemplos de receptividades associadas às transições.

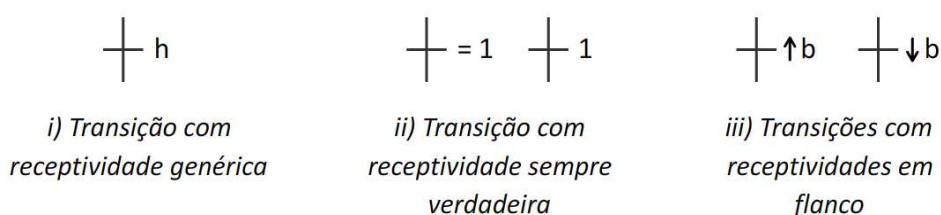


Figura 2.3 – Representação de transições e receptividades

Transição com receptividade sempre verdadeira: é apresentada na Figura 2.3 ii) a notação clássica e alterada (IEC 60848).

Transição com receptividade em flanco: na Figura 2.3 iii), são ilustrados exemplos de receptividade em flanco ascendente (mudança do valor lógico 0 para o valor lógico 1) e descendente (mudança do valor lógico 1 para o valor lógico 0) da variável b .

Na Figura 2.4 são apresentadas receptividades dependentes do tempo. A notação formal $t1/a/t2$ indica que a receptividade só é verdadeira após um tempo $t1$ depois da mudança de estado lógico de "0" para "1" e volta novamente a ser falsa após um tempo $t2$, momento em que a receptividade transita do valor lógico "1" para "0".

⁴Representação utilizada para identificar as etapas a activar nesta situação.

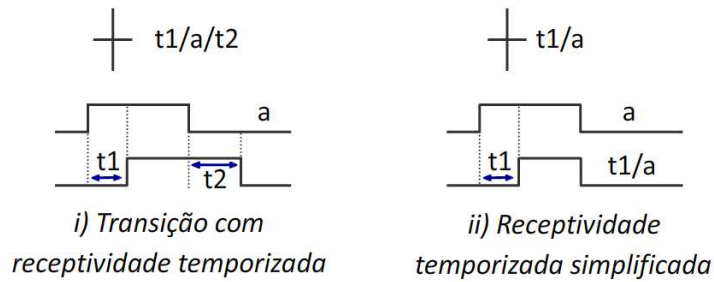


Figura 2.4 – Transições com receptividade temporizada.

Transição com receptividade temporizada: na Figura 2.4 i) a variável de entrada, 'a', deve permanecer verdadeira durante um tempo igual ou superior a $t1$ para que a receptividade associada possa ser verdadeira.

Transição com receptividade temporizada simplificada: é usual considerar receptividades com a temporização $t2$ com o valor nulo - Figura 2.4 ii).

Um *predicado* é uma expressão lógica que contém uma ou mais variáveis. Por predicado, entende-se expressões que podem envolver a utilização de condições lógicas relacionadas, por exemplo, com comparações de variáveis e temporizadores - Figura 2.5.

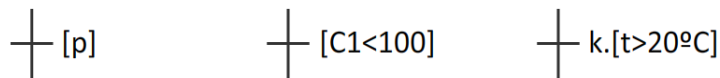


Figura 2.5 – Transição com predicado.

É importante realçar que **formalmente**, duas receptividades baseadas na utilização de flancos, que sejam eventos não correlacionados, não podem ocorrer simultaneamente. Este conceito pode ser importante na escolha de uma sequência, como ilustra a Figura 2.6.

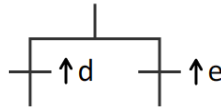


Figura 2.6 – Não simultaneidade da receptividade em flancos.

2.3.3 Acções

As acções permitem estabelecer a ligação entre a evolução do GRAFCET e as saídas do sistema [15]. A nova norma IEC 60848 define dois tipos de acção: **contínuas** e **memorizadas**.

Acção Contínua

A acção contínua é executada enquanto a etapa a que está associada estiver activa. A sua representação é ilustrada na Figura 2.7.

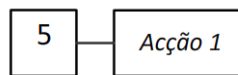


Figura 2.7 – Representação de uma acção contínua.

A execução de uma acção pode estar dependente de uma condição, como se ilustra na Figura 2.8. Essa condição é uma expressão lógica. No caso apresentado na Figura 2.8 i), a acção é executada quando a etapa a que está associada está activa e a condição é verdadeira. Esta condição pode ainda estar dependente do tempo, como se ilustra no exemplo da Figura 2.8 ii). Em ambos os casos a condição nunca poderá ser um flanco de uma variável.

A acção pode ser retardada no tempo (ver Figura 2.9 i)). Só é executada após a activação da etapa associada e decorrido um tempo $t1$. A execução da acção pode também estar limitada no tempo (ver Figura 2.9 ii)). Nesta condição, a acção é executada durante um tempo $t1$ após a activação da etapa associada.

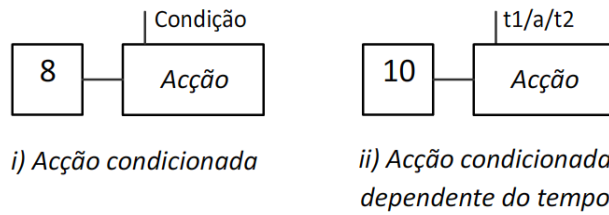


Figura 2.8 – Representação de acções condicionadas.

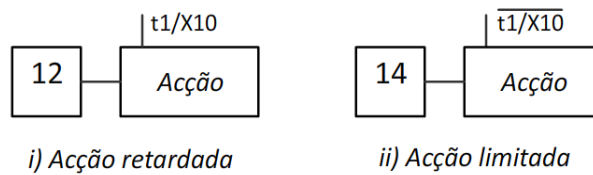


Figura 2.9 – Acção retardada e acção limitada no tempo.

Acção Memorizada

Numa acção memorizada existe uma associação entre a ocorrência de eventos internos e a execução de acções. Um evento é uma expressão lógica que inclui o flanco de variáveis. No exemplo apresentado na Figura 2.10, assim que se verifique o evento, com a etapa associada activa, a acção é executada e memorizada. O seu valor só será alterado pela ocorrência de um novo evento.

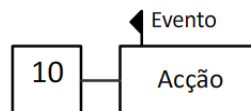


Figura 2.10 – Acção no evento.

No exemplo apresentado na Figura 2.11, a expressão lógica $X2.\uparrow d$ descreve um evento interno em que é atribuída à variável D o valor lógico 1 (quando a etapa 10 está activa).

As acções podem ser activas na activação ou activas na desactivação da etapa correspondente.

Nos exemplos apresentados na Figura 2.12 temos dois casos:

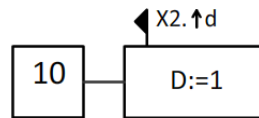


Figura 2.11 – Exemplo de uma acção no evento.

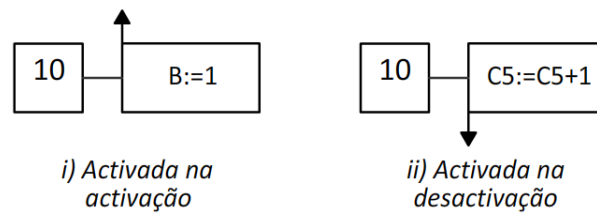


Figura 2.12 – Exemplos de acções memorizadas.

i) **Acção activada na activação:** quando a etapa 10 é activada é atribuído o valor 1 à variável B.

ii) **Acção activada na desactivação:** no momento que a etapa 10 é desactivada, a variável associada ao contador 5, C5, é incrementada.

Outro tipo de acção é a acção no disparo, ilustrada na Figura 2.13. Esta acção é executada no instante do disparo da transição.



Figura 2.13 – Acção no disparo.

2.3.4 Regras de Evolução

As etapas activas num determinado instante são definidas como uma *situação* do Grafcet. Esta *situação* corresponde a um estado do sistema. A evolução entre *situações* do Grafcet rege-se segundo cinco regras [16]:

Regra 1: Estado Inicial

O estado inicial corresponde ao conjunto das etapas activas no início do funcionamento do automatismo. Essa situação é definida pelo responsável do projecto.

Regra 2: Disparo de uma transição

Uma transição está validada quando todas as etapas a montante estão activas. Quando a transição está validada e a respectiva receptividade é verdadeira, a transição é disparada.

Regra 3: Evolução das etapas activas

O disparo de uma transição implica a desactivação de todas as etapas que a precedem e a activação de todas as etapas que a sucedem.

Regra 4: Evoluções simultâneas

Várias transições simultaneamente disparáveis são disparadas em simultâneo.

Regra 5: Activação e desactivação simultânea de uma etapa

No caso de uma etapa ser desactivada e activada em simultâneo, a etapa vai continuar activa (prioridade da activação sobre a desactivação).

2.3.5 Estruturas de base

Sequência e ciclo de uma só sequência

Uma sequência é composta por uma sucessão de etapas que podem ser activas sucessivamente. Neste tipo de sequência, é associada apenas uma transição a jusante de cada etapa. E a montante de cada etapa vai existir apenas uma transição, excepto a primeira [17].

Um ciclo de uma só sequência é uma sequência fechada em que cada etapa tem apenas uma transição associada a jusante. As transições são validadas por uma só etapa da sequência. Na Figura 2.14 são ilustrados estes dois tipos de sequências.

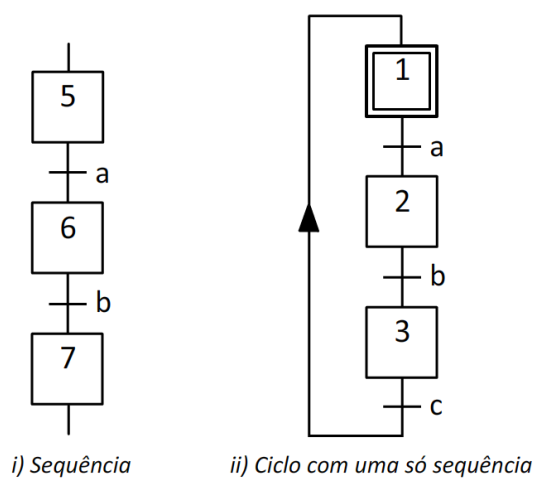


Figura 2.14 – Exemplo de sequência e de um ciclo com uma só sequência.

Seleção e reagrupamento de sequências

A seleção de sequência corresponde a uma divergência OU que permite uma escolha de evolução entre várias sequências a partir de uma ou várias etapas (ver Figura 2.15 i)). Esta escolha de evolução pode ser exclusiva em função das receptividades associadas às transições.

O reagrupamento de sequências corresponde a uma convergência OU que permite, após uma seleção de sequência, continuar por uma sequência única (ver Figura 2.15 ii)).

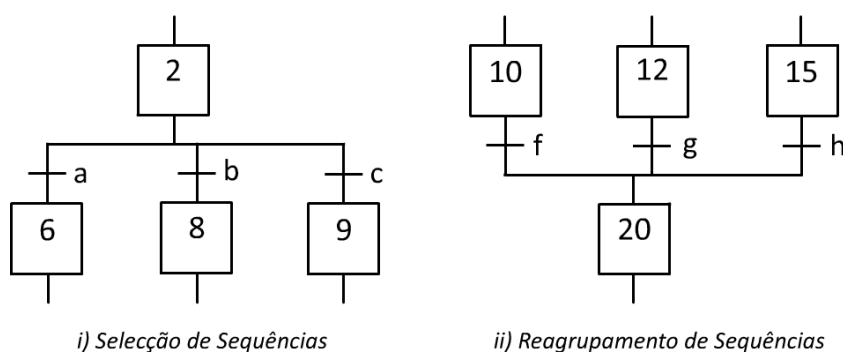


Figura 2.15 – Exemplo de seleção de sequência e reagrupamento de sequências.

Activação de sequências paralelas

É uma situação em que várias sequências são activadas simultaneamente por uma única transição (divergência E), como se exemplifica na Figura 2.16. As diferentes sequências paralelas são totalmente independentes umas das outras e vão ser executadas em simultâneo.

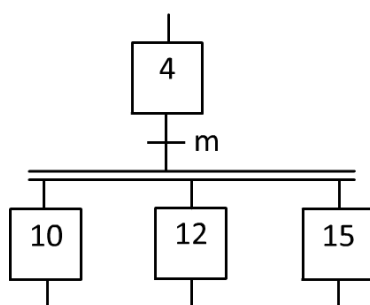


Figura 2.16 – Activação de sequências paralelas.

Sincronização de sequências

A sincronização de sequências é representada por um duplo traço horizontal onde existe uma espera no final das sequências a montante antes de continuar para a sequência posterior (ver Figura 2.17). A transição associada à sincronização é válida (regra nº 2) quando todas as etapas a montante estão activas.

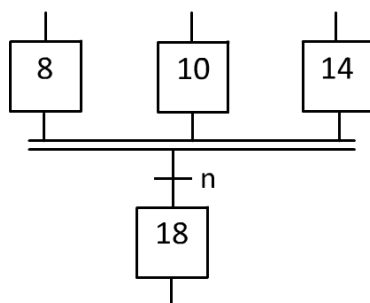


Figura 2.17 – Sincronização de sequências.

Sincronização e activação de sequências paralelas

Neste tipo de estrutura é feita a sincronização de todas as sequências a montante da transição e a activação das várias sequências a jusante da transição, como se exemplifica na Figura 2.18. Esta estrutura também é designada por junção e divergência E.

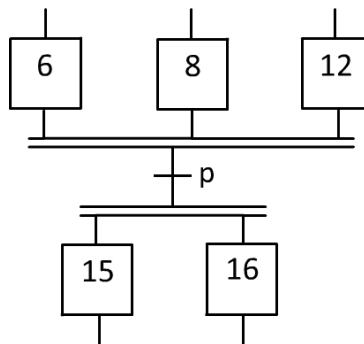


Figura 2.18 – Sincronização e activação de sequências paralelas.

Salto de etapas

O salto de etapas é um caso particular de selecção de sequência que permite percorrer totalmente uma sequência ou saltar uma ou mais etapas. Para um funcionamento correcto as receptividades associadas às transições desta selecção devem ser mutuamente exclusivas.

No exemplo apresentado na Figura 2.19, após a activação da etapa inicial 1, a etapa seguinte será a 4 ou a 2, dependendo do valor das variáveis a e b .

Retoma de sequência

A retoma de sequência é um caso particular de selecção de sequência que permite recomeçar uma mesma sequência. Para um funcionamento correcto as receptividades associadas às transições desta selecção devem ser mutuamente exclusivas. No exemplo apresentado na Figura 2.20, após a activação da etapa 8, a etapa seguinte será a 7 ou a 9, dependendo do valor das variáveis m e p .

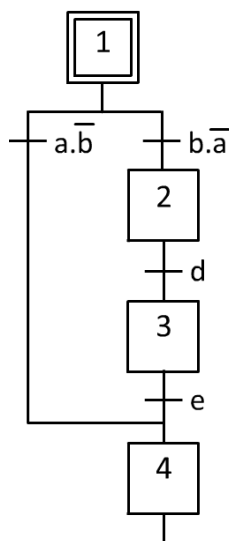


Figura 2.19 – Salto de etapas.

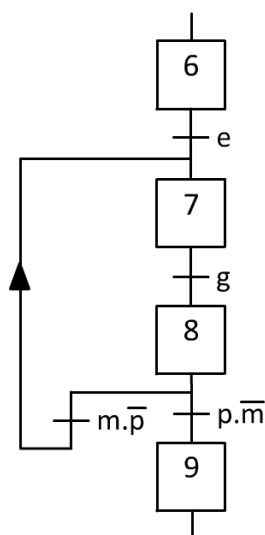


Figura 2.20 – Retoma de sequência.

2.3.6 Estruturação e hierarquia do Grafcet

Forçagem

A forçagem cria uma dependência e uma hierarquia entre vários Grafcets que controlam um mesmo automatismo. Podem ser denominados *Grafcet mestre* e *Grafcet escravo*. A forçagem é uma ordem interna e prioritária sobre as regras de evolução. Os Grafcets forçados são mantidos na situação imposta enquanto a

ordem de forçagem for válida.

As ordens de forçagem são representadas por um duplo rectângulo e indicados os Grafjets parciais de nível hierárquico inferior nos quais a forçagem tem efeito. Algumas representações de forçagem são mostradas na Figura 2.21.

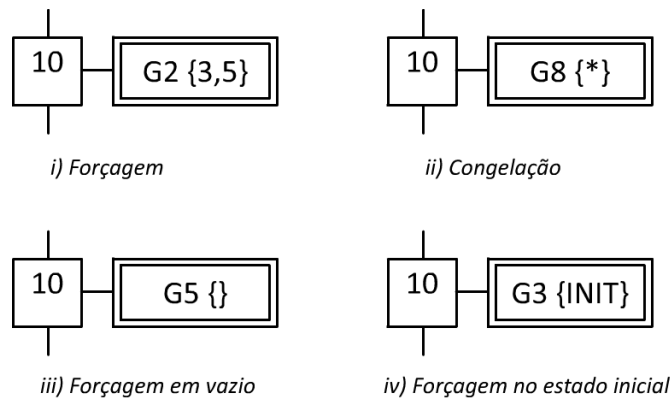


Figura 2.21 – Exemplos de forçagem.

Nos exemplos apresentados na Figura 2.21, quando (e enquanto) a etapa 10 estiver activa, no caso:

- i)** o Grafjet parcial 2 é forçado de forma que as respectivas etapas 3 e 5 fiquem activas. Todas as outras são desactivadas.
- ii)** o Grafjet parcial 8 fica congelado na situação corrente.
- iii)** no Grafjet parcial 5 são desactivadas todas as etapas (situação *vazia*).
- iv)** o Grafjet parcial 3 é colocado no seu estado inicial.

Encapsulamento

Um conjunto de etapas (Grafjet parcial) pode ser encapsulada por uma etapa encapsulante. Desta forma, a estruturação de um Grafjet global pode ser realizada por várias etapas encapsulantes e de maneira hierárquica.

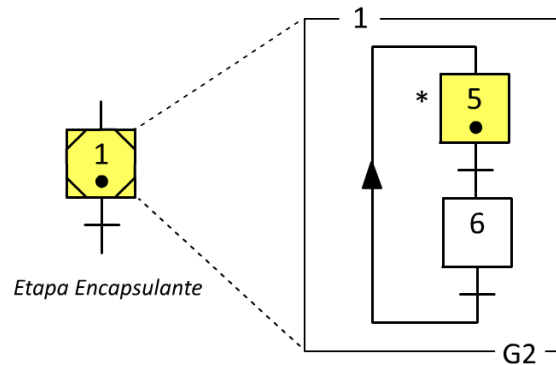


Figura 2.22 – Encapsulamento.

No exemplo da Figura 2.22 o asterisco colocado ao lado da etapa 5 do Grafcet parcial 1, indica que essa etapa será activa (estado inicial de G2) quando a etapa encapsulante 1 for activada. A partir desse momento o Grafcet G2 pode evoluir livremente. A desactivação da etapa encapsulante desactiva todas as etapas encapsulantes e Grafcets parciais a ela associados.

Macro-Etapa

Uma macro-etapa é uma representação única de uma sucessão de etapas e transições [18]. Não pode ser chamada mais do que uma vez. O uso de macro-etapas permite uma melhor representação e compreensão progressiva (descendente) do Grafcet, podendo a expansão da macro-etapa conter uma ou mais macro-etapas.

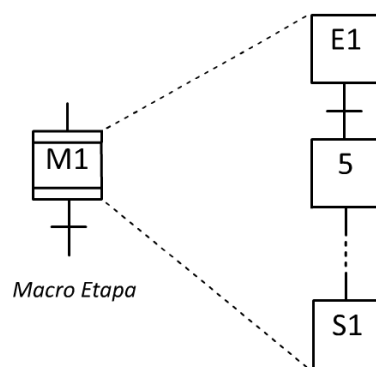


Figura 2.23 – Representação de uma macro-etapa e macro-expansão.

No exemplo da Figura 2.23 está representada uma macro-etapa. No Grafcet da

expansão da macro-etapa 1 é definida a etapa de entrada E1 e a etapa de saída S1 com os respectivos símbolos (E - entrada, S - Saída). A transição a jusante da macro-etapa M1 está validada quando a etapa S1 de saída da expansão está activa.

2.3.7 Outras Estruturas

Etapa Fonte e Poço

A etapa fonte é uma etapa que não tem nenhuma transição a montante. Exemplos de etapa fonte são as etapas iniciais (ver exemplo apresentado na Figura 2.24 i)) e etapas activas por uma ordem de forçagem exercida por um Grafcet hierarquicamente superior, ou por encapsulamento.

A etapa poço é uma etapa que não tem nenhuma transição a jusante, como se pode observar pela Figura 2.24 ii). Para desactivar uma etapa poço é necessária uma ordem de forçagem (proveniente de um Grafcet de nível hierárquico superior) ou então desactivar a etapa encapsulante, caso a etapa poço seja encapsulada.

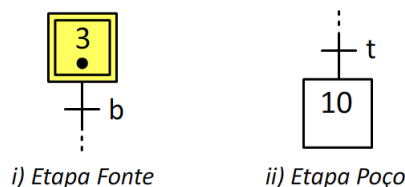


Figura 2.24 – Etapa fonte e etapa poço.

Transição Fonte e Poço

Uma transição fonte é uma transição que não tem nenhuma etapa a montante, (ver Figura 2.25 i)). A transição está sempre validada à qual geralmente se associa uma receptividade com flanco ascendente para evitar disparos sucessivos.

Uma transição poço não tem nenhuma etapa a jusante, (ver Figura 2.25 ii)). O disparo desta transição tem por objectivo a desactivação das etapas a montante. De igual forma que a transição fonte, é prudente associar uma receptividade com flanco ascendente a este tipo de transição.

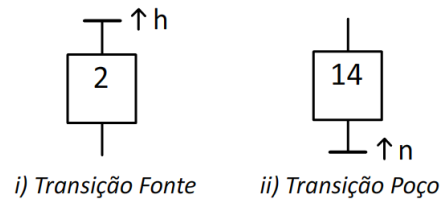


Figura 2.25 – Transição fonte e transição poço.

2.4 Editores de Grafcet

Nesta secção são apresentados alguns editores que permitem o desenho e programação em GRAFCET. Um exemplo de editor gráfico de GRAFCET é descrito em [19], onde a ferramenta desenvolvida facilita o desenho de GRAFCET permitindo a visualização dos programas com uma interface amigável.

2.4.1 SFCEdit

SFCEDIT [20] é um *software* para modelar o GRAFCET de acordo com o padrão do IEC 60848. Este programa mantém a disposição dos elementos na área de trabalho de forma automática e auxilia nos passos seguintes do desenho o que torna a manipulação dos elementos uma tarefa mais fácil. Estas representações são ideais para os sistemas de evolução sequencial pois permitem visualizar o desenrolar dos automatismos decompondo-o em diversas etapas. Esta ferramenta contém todos os componentes gráficos necessários para projectar as mais variadas estruturas em GRAFCET, como por exemplo: estruturação por macro-etapas ou por etapas encapsulantes. SFCEDIT é um instrumento útil para conceber Grafkets complexos para a documentação do projecto de automatismos (ver Figura 2.26).

2.4.2 GRAFTOR

GRAFTOR [21] é um programa para todos os engenheiros e profissionais que pretendam conceber sistemas automatizados. Este programa corre na plataforma Linux. A interface gráfica (ver Figura 2.27) permite o desenho do GRAFCET de

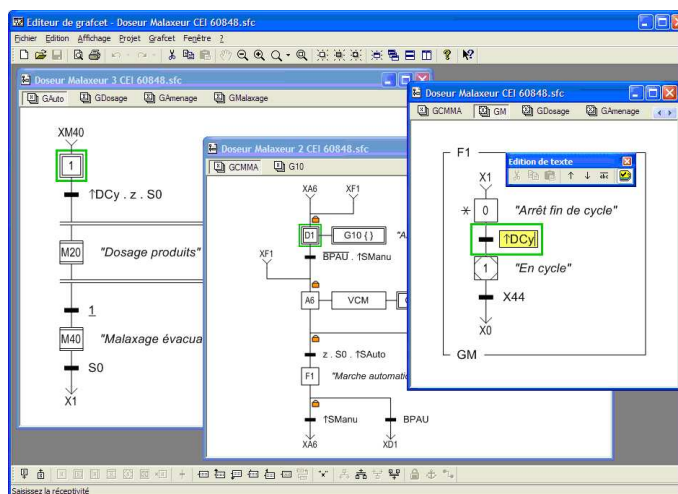


Figura 2.26 – Ambiente de trabalho do SFCEdit.

uma maneira simples e fácil. A ligação entre as etapas e as transições é feita de maneira automática. É dada a possibilidade de imprimir o trabalho elaborado ou exportar para o formato Postscript.

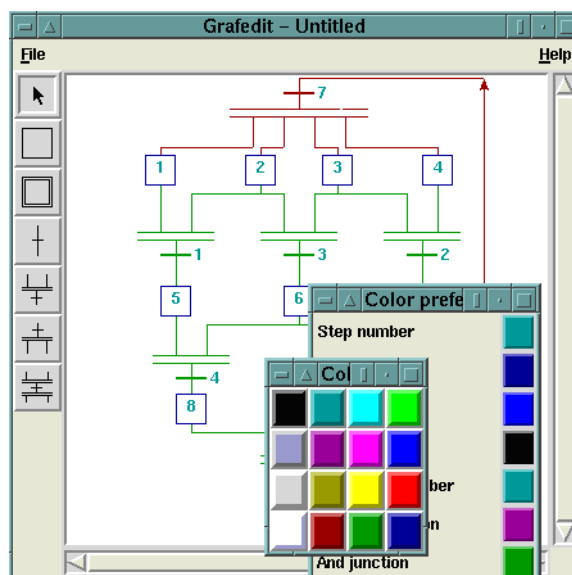


Figura 2.27 – Exemplo de janela do Graftor.

2.4.3 EdiTSAB

O EdiTSAB [22] é um *software* simples e amigável que, para além de realizar esquemas eléctricos, pneumáticos e hidráulicos, permite uma edição rápida do GRAFCET. Este editor de GRAFCET está em conformidade com a norma EN 60848 (Figura 2.28). O *software* EdiTSAB foi desenvolvido para os estudantes do liceu Antoine Bourdelle em Montauban e pode ser utilizado e difundido livremente.

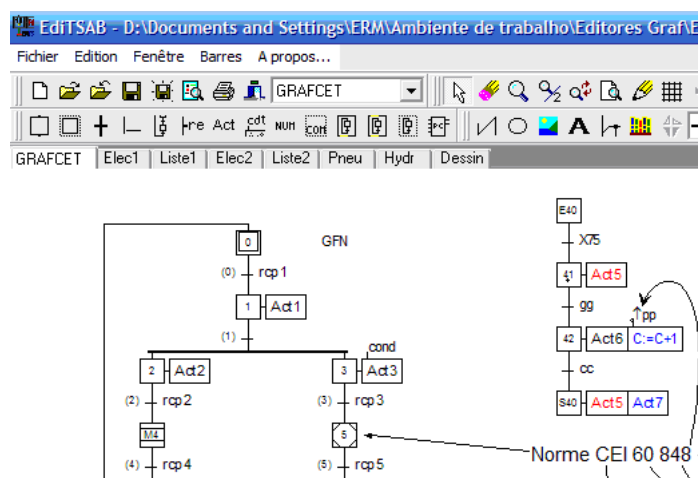


Figura 2.28 – Ilustração retirada do EdiTSAB.

2.4.4 MACHCET

MACHCET [23] é um *software* interactivo que se destina à iniciação aos automatismos no que concerne à linguagem GRAFCET. Utiliza um conceito lúdico de realização de máquinas virtuais, conservando ao mesmo tempo o formalismo dos automatismos. O MACHCET permite abordar todas as fases de concepção de um automatismo, desde a concepção da máquina à elaboração do programa de comando em linguagem GRAFCET.

O *software* é constituído por um editor de máquina virtual, uma biblioteca de actuadores e sensores e uma biblioteca de peças que, no seu conjunto, vai permitir conceber a construção da máquina/animação que se pretende realizar.

MACHCET é um editor muito bom que permite gerar o programa Grafcet da situação concebida (ver Figura 2.29). Para além do compilador GRAFCET, este *software* contém um analisador de estrutura GRAFCET, para garantir o respeito da norma e funcionamento.

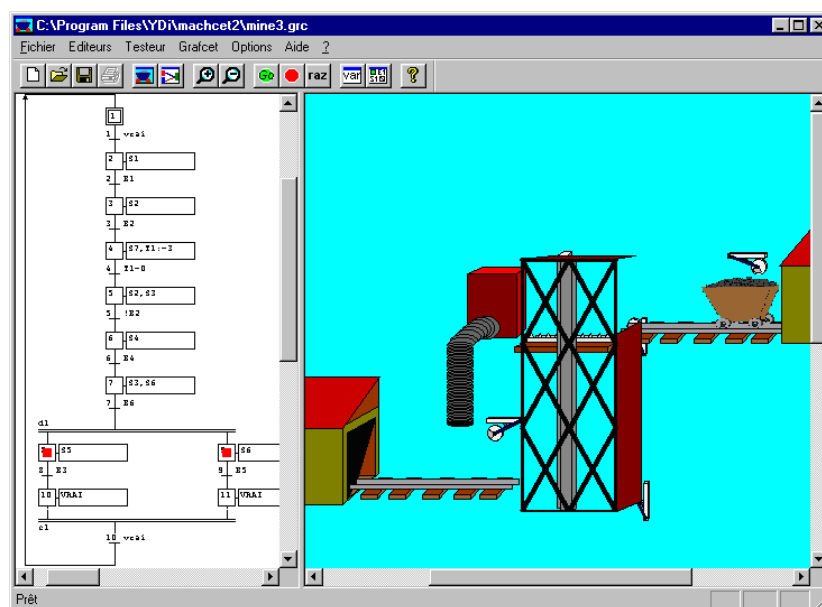


Figura 2.29 – Exemplo retirado do MACHCET.

2.4.5 WinGrafcet

O WinGrafcet [24] é um *software* de desenho e simulação de processos industriais pelo uso da linguagem Grafcet. A disposição dos vários elementos de desenho no ambiente de trabalho do WinGrafcet, como se pode observar pela Figura 2.30, facilita a elaboração dos programas em Grafcet.

2.4.6 Grafcet

Este programa foi desenvolvido por Montero Ribas [25] permite desenhar os Grafcets e guardar o resultado no formato vectorial (.EMF e .SVG) ou em imagem (.BMP e .JPG). É relativamente fácil e intuitivo o desenho do GRAFCET e a definição de todos os elementos que o constituem (Figura 2.31).

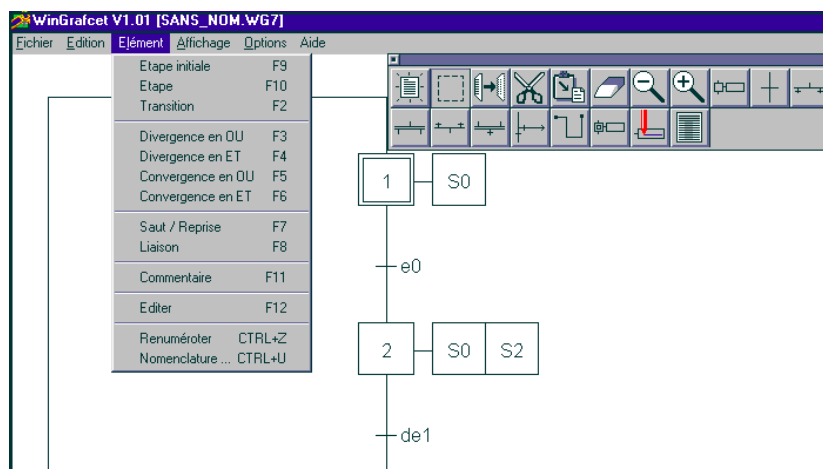


Figura 2.30 – Exemplo do WinGrafcet.

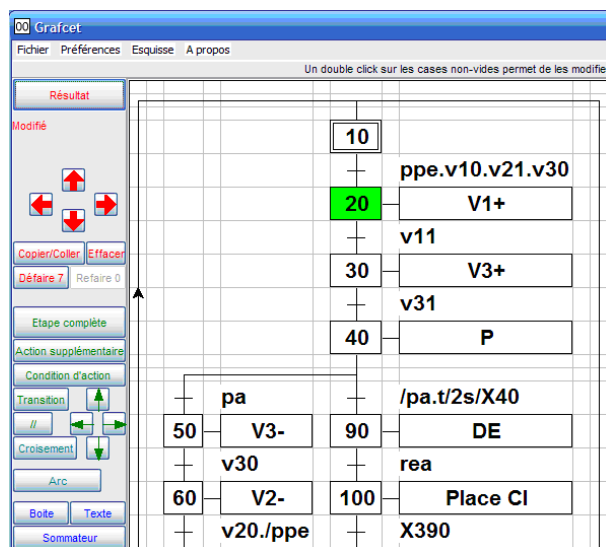


Figura 2.31 – Ambiente de trabalho do Grafcet.

2.5 Software de Automação

No dia-a-dia, quando trabalhamos com PLC, estes têm de ser programados de alguma forma. Uma grande parte dos fabricantes de PLC inclui o Grafcet como linguagem de programação, nas suas versões mais recentes.

A grande maioria do *software* de automação permite a programação do PLC através

de diagrama de contactos (*Ladder Diagrams*). Para estes casos, o Grafcet pode ser convertido em equações lógicas e representado nestes diagramas.

A possibilidade dada pelo *software* de automação de animar graficamente os SFC é bastante importante no processo de trabalho especialmente se forem encontrados erros de concepção pelo módulo de análise.

Na secção seguinte seguida é listado *software* de automação que permite escrever programas em SFC.

2.5.1 Automation Studio

O Automation Studio [26] é um *software* que engloba um pacote completo e integrado que inclui várias tecnologias de automação, que permite ao utilizador desenhar e simular (com animação) os sistemas de automatização projectados [27]. Na Figura 2.32 é apresentada uma ilustração do ambiente de programação do Automation Studio.

O módulo SFC dispõe de uma biblioteca extensa de tipos de componentes para a implementação de estruturas de controlo de acordo com a norma IEC 61131-3. Uma grande vantagem do Automation Studio é permitir ao utilizador criar e armazenar os seus próprios símbolos e bibliotecas de modo a fazer face às especificações pretendidas, usando para o efeito os componentes padrão, a flexibilidade das ferramentas e os grupos de funções.

O Automation Studio contém um kit de interface I/O (opcional) para ligação a dispositivos externos reais. Assim, pode-se usar o Automation Studio como um simulador SoftPLC para controlar sistemas reais tais como painéis didácticos, pneumáticos ou eléctricos. Com este kit, os estudantes podem desenhar e simular qualquer processo controlado por um PLC real, usando elementos das bibliotecas e transformar o Automation Studio numa completa fábrica virtual. Este módulo permite o controlo mais eficiente de sistemas pneumáticos, hidráulicos ou eléctricos e fornece um bom suplemento para a documentação do projecto.

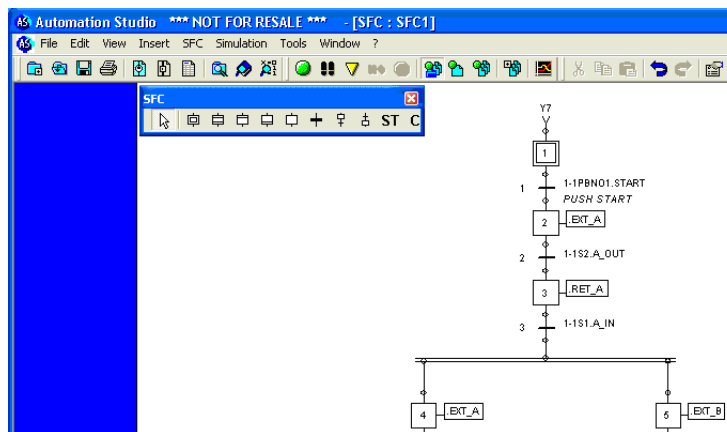


Figura 2.32 – Ilustração do Automation Studio.

2.5.2 CoDeSys

Com o *software* CoDeSys [28] podemos programar em todos os editores definidos no padrão do IEC para o desenvolvimento de uma aplicação: Texto Estruturado (ST), Sequential Function Chart (SFC), Diagrama de blocos de funções (FBD), Diagrama de contactos (LD) e Lista de Instruções (IL).

Os editores foram todos projectados de modo a assegurar uma óptima manipulação dos elementos e é possível alternar com facilidade entre os vários editores durante a execução do projecto. Os elementos podem ser incorporados directamente no editor ou arrastados de uma caixa de ferramentas (ver Figura 2.33). O editor de SFC pode ser usado como definido no padrão ou numa versão simplificada. Este editor de SFC contém ainda uma ferramenta que permite fazer uma análise no tempo das etapas de uma forma confortável. As etapas e as transições assim como os restantes elementos presentes neste editor podem ser encapsulados em macros.

2.5.3 IsaGRAF

ISaGRAF [29] é um ambiente de *software* de controlo que permite criar sistemas de controlo locais ou distribuídos. Oferece uma combinação de um motor de controlo robusto (máquina virtual) e de um ambiente intuitivo no desenvolvimento da aplicação.

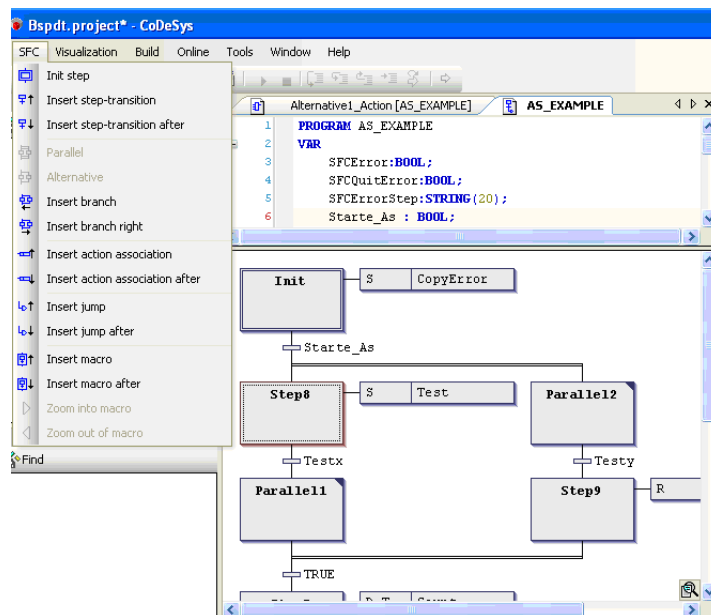


Figura 2.33 – Ambiente de trabalho do CoDeSys.

Na linguagem SFC (de acordo com o padrão do IEC 61131-3), o ciclo do processo é dividido num número de etapas bem definidas, separadas por transições. As outras linguagens são usadas para descrever as acções executadas dentro das etapas e das condições lógicas para as transições. Os processos paralelos podem facilmente ser descritos pelo uso desta linguagem (SFC).

O editor de SFC de ISaGRAF fornece características de edição eficientes e fáceis de usar, como rato e teclado direccionado. A interface gráfica e o uso de cores nos vários elementos permite ao utilizador inserir, seleccionar e movimentar esses elementos de uma forma muito agradável (ver Figura 2.34). Durante o desenvolvimento o SFC é reajustado automaticamente de modo a proporcionar uma imagem e organização claras. O nível 2 do SFC, que é o programa a ser executado numa determinada etapa, é indicado numa janela deslizante. Contém ainda uma galeria de SFC que permite a inserção rápida dos elementos e estruturas complexas.

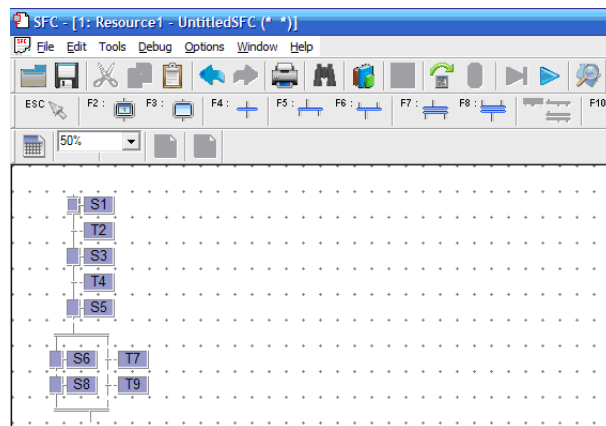


Figura 2.34 – Ilustração do IsaGRAF.

2.5.4 STRATON Workbench

STRATON [30] é um *software* com suporte para as cinco linguagens de programação presentes na norma IEC 61131-3. O ambiente de desenvolvimento com uma única janela e uma barra de ferramentas simples permite a adaptação fácil para o utilizador. Faz uso da tecnologia "Drag and Drop" e tem uma boa ajuda em linha (*online*), (ver exemplo da Figura 2.35).

STRATON inclui um compilador com elevado desempenho que fornece, como opção, diferentes tipos de código. Gera automaticamente um ficheiro HTML que serve de documentação ao projecto. São fornecidas várias ferramentas úteis, no modo em linha, com a finalidade de eliminar possíveis erros no decorrer do projecto.

2.5.5 PL7

O *software* PL7 foi projectado para desenvolver aplicações de sistemas de controlo. Permite fazer modificações ao programa no modo em linha (*online*) e tem ferramentas que ajudam na detecção de erros. O programa pode ser feito juntamente com animação, o que proporciona uma melhor compreensão e optimização do projecto a ser realizado.

O acesso rápido aos serviços disponíveis, quando se selecciona um objecto com o

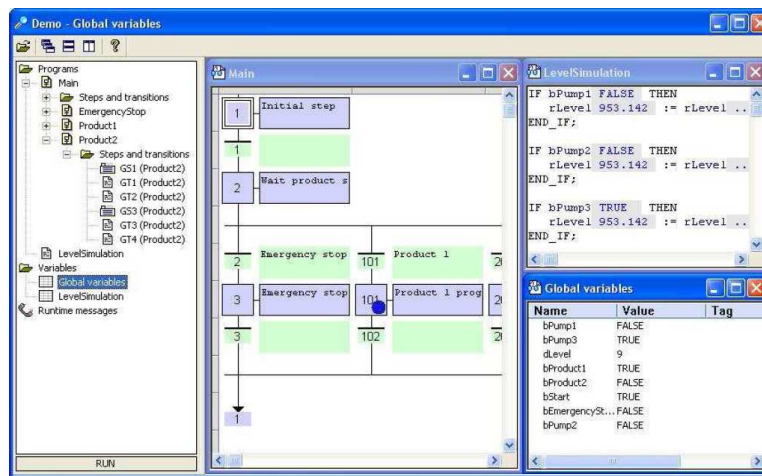


Figura 2.35 – Ilustração do STRATON.

rato e a sua respectiva ajuda, aliados ao ambiente de trabalho amigável torna o processo mais produtivo (ver Figura 2.36).

Neste *software*, os programas escritos na linguagem SFC consistem em: macro-etapas - que é uma representação de uma sucessão de etapas e transições; etapas - onde estão associadas as acções a ser executadas; transições - são associadas condições de transição; ligações orientadas - fazem a ligação das etapas às transições.

As acções (contínuas, pulsadas na activação ou na desactivação) e as condições da transição podem ser programadas na linguagem desejada: Texto Estruturado (ST), Diagrama de contactos (LD) e Lista de Instruções (IL).

2.6 Sítios e ferramentas que permitem aprender o GRAFCET

Nesta secção são descritos alguns trabalhos similares e sítios existentes na Internet, de maior relevo, com o objectivo de melhorar a aprendizagem do GRAFCET.

Em [31] é apresentado um curso com os conceitos básicos do GRAFCET, onde são propostos exercícios e exemplos de automação através de simulações interactivas. O curso está dividido em vários pacotes de informação e no final de cada pacote o

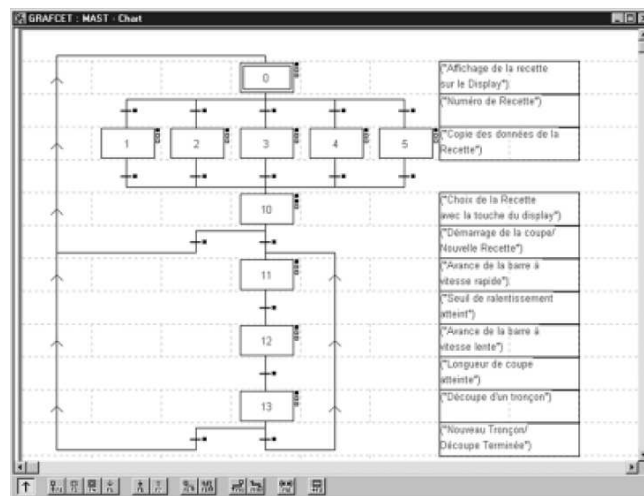


Figura 2.36 – Ambiente de trabalho do PL7.

aluno pode fazer uma auto-avaliação dos conhecimentos adquiridos. Estes pacotes estão ordenados por ordem de complexidade e são constituídos por: uma introdução teórica; perguntas e exemplos da implementação do Grafcet em controladores lógicos; finalmente são apresentadas simulações que representam processos a ser automatizados, onde o aluno tem de desenvolver o Grafcet de controlo para essa simulação e enviar a sua resolução ao professor.

Um curso similar desenvolvido por Francisco Santos pode ser encontrado em [32]. Neste curso multimédia que está adaptado para incapacitados visuais, é reproduzido (pelo autor) o áudio de todos os conteúdos teóricos. O curso está dividido em quatro partes: - introdução ao GRAFCET: onde são descritos os elementos e estruturas necessários para compreender e escrever programas em Grafcet; - implementação do GRAFCET: descreve como se pode obter as equações lógicas do Grafcet as quais descrevem o funcionamento do automatismo; - desenho estruturado: é demonstrada a necessidade de estruturação do Grafcet, os vários modos de funcionamento e como implementar mecanismos de segurança para pessoas e instalações; - exercícios: são apresentados alguns exemplos com a proposta de resolução em Grafcet e as equações que representam as etapas e as acções.

Na Universidade do Minho foram realizadas três experiências com a aplicação do

conceito de 'blended-learning' [33, 34]. Na disciplina de Automação foi projectada uma página *web* [35] como ferramenta de ajuda para os alunos. Essencialmente, tem disponível o conteúdo teórico da disciplina, algumas animações de simulações e é possível avaliar os conhecimentos adquiridos pela realização de testes em linha [36]. As animações apresentadas estão bem representadas e simulam situações reais. A modelação destes exemplos em Grafcet é um pouco restritiva, na medida em que são representadas apenas algumas das suas estruturas. Para além desta experiência, um grupo de alunos do 4º ano desse mesmo curso, desenvolveu kits didácticos que simulam ambientes reais: controlo de trânsito por semáforos, uma pequena casa inteligente e um sistema de gestão de uma fila de competição [6].

Francesc Estragués [37] apresenta uma descrição do GRAFCET e as várias estruturas (sem recurso a animação). São enunciados exemplos práticos, alguns deles com a resolução em Grafcet e a sua implementação, mas nem em todos é possível observar a evolução do Grafcet com o decorrer da animação.

O sítio de Patrick Trau [38] contém uma descrição das partes essenciais do GRAFCET. São apresentados três exemplos práticos com resolução, que representam algumas estruturas (prioridade, sequência e semáforo).

Na página realizada por Philippe Berger [39] está disponível uma introdução e definição teórica dos elementos e estruturas do GRAFCET. De igual forma, Patrick Abati [40] elaborou uma página onde descreve sucintamente, com exemplos, os elementos que constituem o GRAFCET, a partir dos quais são elaborados os programas. Em [41] é apresentado um breve curso de GRAFCET com algumas animações.

Robert Papanicola [42] propõe uma descrição do GRAFCET em duas partes. Na primeira parte descreve as regras e os elementos que o constituem, e na segunda parte descreve a estruturação e a hierarquização.

Em [43] é permitido o acesso a um curso completamente teórico, segundo a norma EN 60848, onde os vários elementos e estruturas são explicados e representados por

imagens. Outro curso teórico bastante similar é apresentado em [44].

Oriol Boix [45] disponibiliza uma descrição teórica do GRAFCET bastante completa com muitos exemplos estáticos e descreve modos de implementação do Grafcet.

Em [46] é descrito um sistema de brocagem automática. Além de uma simples descrição do GRAFCET, contém uma componente interactiva onde é proposta a resolução de um problema com base num menu de escolhas para as etapas e transições. É uma boa ideia, contudo o seu funcionamento pode ser melhorado e a animação, na qual se pretende descrever o controlo, deve estar presente na página de interacção.

Em [47] é apresentado um curso bastante completo com imagens a representar as estruturas e os vários exemplos. No final é proposta a resolução de testes e a realização de trabalhos práticos em laboratório.

2.7 Interesse deste tipo de ferramentas

Ao longo dos anos observa-se um aumento de informação disponibilizada na internet sobre o GRAFCET. No entanto, esta informação é na maioria dos casos transmitida na forma textual e apenas com algumas animações, ou então, é de uso interno nos locais onde é desenvolvida.

Interessante seria, dispor de uma ferramenta capaz de abranger os vários conteúdos referentes ao GRAFCET e transmitir toda essa informação pelo intermédio de simulações, onde os conhecimentos teóricos aliados a demonstrações práticas permitam uma maior auto-aprendizagem por parte dos alunos. Será também interessante: o uso desta ferramenta no apoio às unidades curriculares de Automação Industrial; a disponibilidade da ferramenta ao maior número de utilizadores possíveis e que esta seja de utilização livre.

Neste cenário, é de enorme interesse o desenvolvimento de uma ferramenta a este nível, capaz de demonstrar a informação com clareza proporcionando aos utilizadores um suporte na aprendizagem do GRAFCET. Este é o objectivo global

deste trabalho.

2.8 Notas Finais

Este capítulo foi dedicado à relevância do GRAFCET como ferramenta de modelação de sistemas de eventos discretos. Assim sendo, foi apresentada uma introdução ao GRAFCET como ferramenta de modelação de processos sequenciais. Neste contexto, foram abordadas editores que permitem o desenho e programação em SFC, bem como soluções existentes no mercado no que concerne a *software* de programação para PLC.

Para finalizar o capítulo foram apresentadas as ferramentas e sítios de Internet existentes para aprender o GRAFCET e referida a necessidade no desenvolvimento de uma nova ferramenta adequada às actuais necessidades.

De acordo com o que foi apresentado, surge então a necessidade de implementar tal ferramenta. Por questões de disponibilidade a Internet é sem dúvida a melhor plataforma de suporte a esta ferramenta, sem excluir a disponibilidade em suporte físico. O capítulo seguinte descreve a metodologia utilizada, as tecnologias que vão permitir a sua implementação e a descrição de todos os conteúdos que a constituem.



e-GR AFCET

3.1 Introdução

Neste capítulo, é apresentada a descrição da ferramenta de apoio ao ensino do GR AFCET designada de e-GR AFCET, bem como as opções tomadas quanto ao software utilizado para desenvolver toda a ferramenta e todo o planeamento a nível do conteúdo, funcionalidade e estética.

O e-GR AFCET é uma ferramenta educacional que permite o apoio e complemento ao ensino do GR AFCET. O nome escolhido para a ferramenta deriva do assunto em que se insere (GR AFCET) e da adaptação da moderna técnica de ensino baseada no *e-learning*. Daí o nome de: e-GR AFCET.

3.2 Estrutura do e-GR AFCET

3.2.1 E-LEARNING

O Tratado de Bolonha vêm mudar o modelo de ensino que tem vindo a ser praticado nas universidades. A diminuição do número de horas presenciais de contacto com o professor leva a uma reformulação das metodologias de ensino/aprendizagem [48].

O e-Learning é ”o processo pelo qual, o aluno aprende através de conteúdos colocados

no computador e/ou Internet e em que o professor, se existir, está à distância utilizando a Internet como meio de comunicação (síncrono ou assíncrono), podendo existir sessões presenciais intermédias.” [49]. A aprendizagem síncrona está associada ao conceito de aprendizagem em tempo real, onde o professor e aluno estão face a face (modelo presencial tradicional) ou o professor conduz todo o processo em linha (modelo presencial à distância). Na aprendizagem assíncrona a interação entre professor/aluno ocorre com um certo intervalo de tempo, onde o aluno determina o seu ritmo de estudo no local que achar mais agradável [50]. Nesta modalidade são utilizados recursos para harmonizar e disponibilizar o conteúdo educativo, como por exemplo o CD-ROM e a Internet [5, 51].

Neste contexto, o e-GRAFCET pretende ser uma ferramenta que combina todos estes meios de aprendizagem, desde o ensino presencial e à distância, recorrendo a tecnologia multimédia para desenvolver a ferramenta, permitindo um ensino mais dinâmico e motivador [52].

3.2.2 Adobe Flash

A tecnologia multimédia interactiva pode ajudar a motivar os alunos transmitindo a informação de uma forma concreta e perceptível [5, 53].

O software escolhido para o desenho e animação dos vários conteúdos foi o Adobe Flash®. É uma ferramenta moderna e flexível que permite conciliar os vários elementos activos da animação: animação propriamente dita do problema proposto, em conjunto com a simulação do Grafcet associado a esse problema e à implementação do mesmo num Controlador Lógico Programável. Todo o controlo da animação é feito com recurso à linguagem de programação ActionScript que é parte integrante do software Flash® [54, 55]. Outra vantagem da utilização desta ferramenta é o reduzido espaço ocupado no servidor por cada uma das animações. É utilizada a tecnologia de compressão existente no Flash, que reduz significativamente o tamanho de cada arquivo. A Internet, vai permitir uma rápida disponibilidade da informação para o utilizador.

Quanto ao grafismo, todos os elementos em cena foram cuidadosamente desenhados para obter um resultado final com qualidade, tanto a nível de imagem como na boa compreensão dos conceitos.

ActionScript

O ActionScript é uma linguagem de programação baseada em ECMAScript ¹, usada para controlar filmes e aplicações do Adobe Flash. O ActionScript 2.0 trabalha com *movie-clips*, campos de texto e som. Esta linguagem torna o ambiente do Flash muito mais interessante de se trabalhar, pois permite desenvolver programas bem elaborados com ambientação gráfica.

A chave para o sucesso das animações em Flash é o recurso aos *movie-clips*, onde os objectos são definidos com animação própria, e pelo uso da linguagem de programação ActionScript, permitindo controlar toda a evolução da cena e interactividade com o utilizador.

3.2.3 Adobe Dreamweaver

O software de desenvolvimento para a construção da página de Internet é o Dreamweaver® da Macromedia. Um dos objectivos iniciais foi definir regras para a construção do sítio, de modo a proporcionar um bom aspecto e uma evolução coerente. A estrutura do *site* foi previamente planeada e a informação que se pretendia colocar acessível foi toda organizada. O grafismo e concepção das páginas foi pensado para serem consistentes e funcionais, o que torna fácil e simples o seu uso. Tudo isto para manter o interesse por parte dos utilizadores.

A página principal está dividida em várias quadros, dois dos quais considerados os quadros principais: o quadro de tópicos, onde se encontra os elementos do menu de opções, e o quadro de informação, onde são carregados e visualizados os dados sobre um determinado tópico. O quadro com um maior destaque contém o logótipo do

¹ECMAScript é uma linguagem de programação baseada em scripts, padronizada pela Ecma International na especificação ECMA-262. A linguagem é bastante usada em tecnologias para Internet, sendo esta base para a criação do JavaScript/JScript e também do ActionScript.

e-GRAFCET e vai aparecer em todas as páginas, como se pode observar na Figura 3.1.

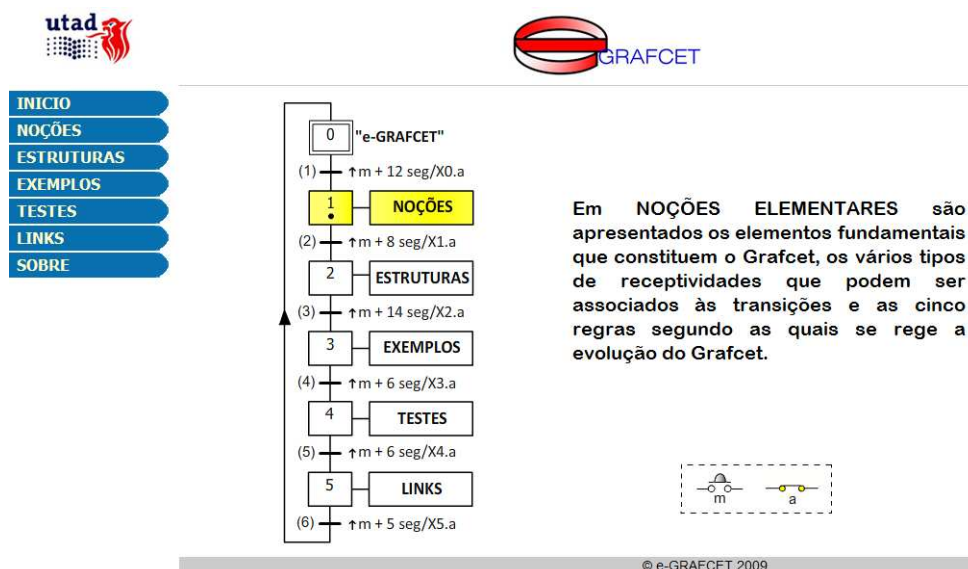


Figura 3.1 – Página inicial do e-GRAFCET.

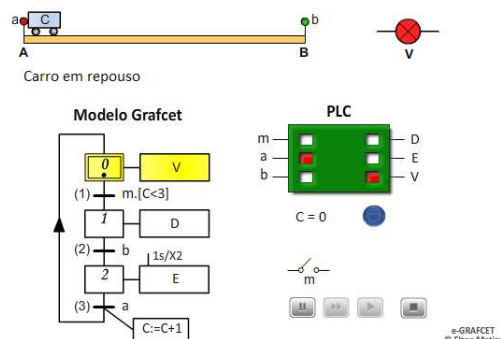
Na página dos vários exercícios didáticos a resolução de cada um é apresentada numa nova página, para que seja possível acompanhar toda a animação sem perder de vista o enunciado do problema. A página de resolução de cada exercício é ainda acompanhada com ajuda ao problema que foca os pontos essenciais dessa simulação, como se pode observar na Figura 3.2.

A qualquer momento e em qualquer página, o utilizador é capaz de aceder com facilidade a todos os elementos do menu.

Outro aspecto importante é o tempo que a página pode demorar a carregar. Sendo esse um factor importante para o sucesso de qualquer página os conteúdos foram estrategicamente divididos por várias páginas. De acordo com o espaço médio ocupado por cada animação Flash foi estabelecido um limite de informação por página de 200KB². Desta forma o utilizador terá de esperar apenas alguns segundos,

²Um KiloByte (KB) é uma unidade de medida usada na informática. Ela mede a quantidade de informação contida num determinado arquivo.

CICLO DE UMA SÓ SEQUÊNCIA - 4



AJUDA – Exemplo 1

Descrição:

- Este exemplo pretende mostrar a evolução do ciclo de uma só sequência.

Pontos Essenciais:

Exemplo 1.1:

- Transição ascendente da variável m ('0' para '1' lógico) inicia a sequência.
- As acções são contínuas e estão associadas às respectivas etapas.

Exemplo 1.2:

- Transição ascendente da variável associada ao interruptor m inicia a sequência.
- O uso do interruptor permite verificar que a sequência apenas volta a ser executada quando este transita de '0' para '1' lógico.

Figura 3.2 – Página de um exemplo prático com ajuda.

para uma velocidade lenta (56kbps) de acesso à Internet. Para as actuais velocidades de acesso, a informação é carregada muito rapidamente, demorando um a dois segundos.

O alojamento de toda a ferramenta, como foi referido no capítulo 2, está disponível em linha (*online*) através de um servidor web e em diferido (*offline*) pelo uso de CD-ROM's.

3.2.4 XML

XML³ é uma recomendação da W3C⁴, para superar as limitações do HTML Hyper-Text Markup Language, que é o padrão das páginas da Web [56].

O XML é uma linguagem de marcação de documentos de texto e de dados, totalmente independente das plataformas de *hardware* e *software* que a utilizam. A marcação é realizada através da utilização de etiquetas de marcação ou palavras-chave (*tags*), muito semelhantes às utilizadas em documentos HTML [57].

Em ambas as linguagens, cada *tag* consiste em duas partes, uma que inicia e outra

³eXtensible Markup Language

⁴World Wide Web Consortium - entidade responsável pela definição da área gráfica da internet

que fecha o comando. No HTML as *tags* estão pré-definidas, enquanto no XML as *tags* são definidas pelo utilizador e podem significar qualquer coisa que se pretende representar.

Pela versatilidade e simplicidade da linguagem XML, esta foi escolhida para representar e guardar os dados referentes às perguntas e respostas para a realização dos testes *online*.

3.3 Descrição da ferramenta

De seguida é apresentada, com mais detalhe, toda a informação que constitui o e-GR AFCET.

3.3.1 Página Inicial

Como referido anteriormente, logo que entramos no *site* é apresentada a página inicial. Esta página é constituída pelo logótipo da ferramenta e-GR AFCET, pelo menu de opções que vai dar acesso a todos os conteúdos e pela animação inicial onde é feita uma breve apresentação do e-GR AFCET e dos conteúdos que podem ser encontrados.

3.3.2 Noções Elementares

Em noções elementares é dado a conhecer os elementos que servem de base para a compreensão do GR AFCET.

Elementos fundamentais do GR AFCET

Aqui são apresentados, passo a passo com a devida descrição, todos os elementos que constituem o GR AFCET. No final é dada a possibilidade de rever toda a apresentação ou apenas relembrar a descrição de algum elemento pela simples passagem do rato sobre ele, Figura 3.3.

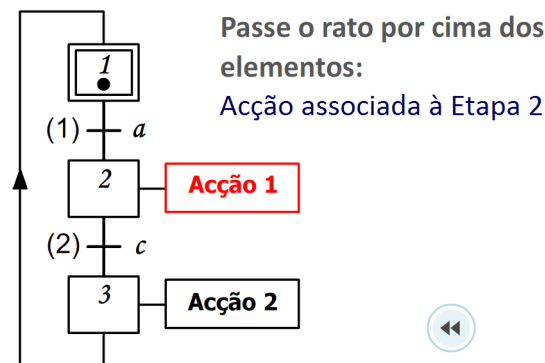


Figura 3.3 – Representação dos elementos fundamentais do GRAFCET.

Receptividades

Uma receptividade é uma função lógica que está associada a cada transição. Na Figura 3.4, são ilustradas os vários tipos de funções que podem ser associadas às transições. Um exemplo é a receptividade sempre verdadeira em que a transição está sempre validada. Outros exemplos são as funções com variáveis lógicas, por exemplo, sinais de entrada (botões, sensores, entre outros) e variáveis internas. Da mesma forma, podem ser associadas variáveis não Booleanas em que estas são válidas quando ocorre uma mudança de estado: transição de 0 para 1 lógico (flanco ascendente) ou de 1 para 0 (flanco descendente). Esta última situação é acompanhada de um exemplo com cronograma para os dois casos.

Regras de evolução

A evolução do Grafcet rege-se por cinco regras. As regras vão ser apresentadas e comentadas uma a uma, acompanhadas com um exemplo ilustrativo. O exemplo da Figura 3.5 corresponde à regra 5 que refere o caso de haver uma activação e desactivação simultânea de uma etapa, onde a prioridade é dada à activação, ou seja, a etapa permanece activa.

Em qualquer momento pode-se avançar para a regra seguinte ou rever as regras anteriores. Também é dada a possibilidade de repetir a explicação da regra em causa.

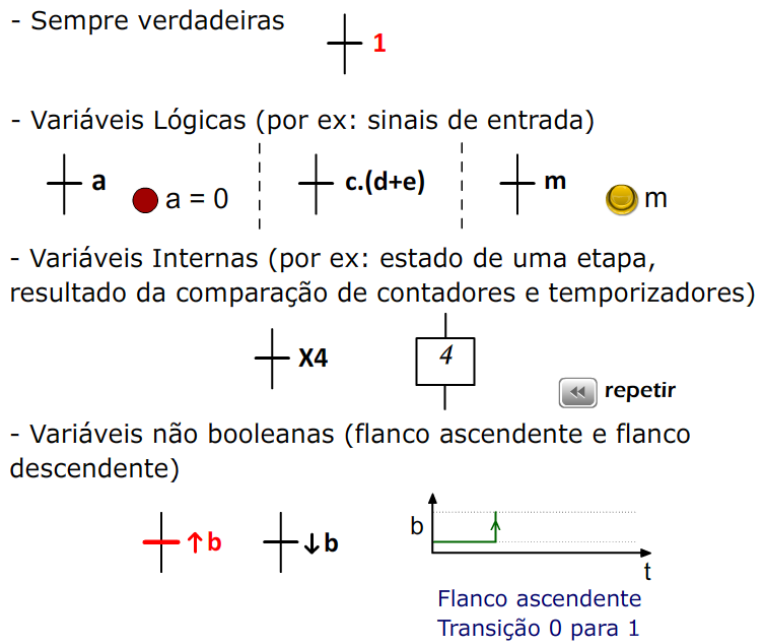


Figura 3.4 – Receptividades associadas às transições.

Regra 5: Activação e desactivação simultânea de uma etapa

- Neste caso a etapa vai permanecer activa (prioridade da activação sobre a desactivação).

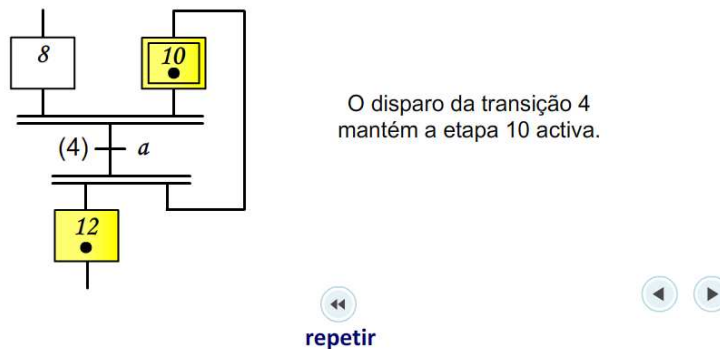


Figura 3.5 – Evolução do Grafcet - Regra 5.

3.3.3 Estruturas Elementares

As estruturas do GRAFCET estão divididas em dois tipos: elementares e avançadas. Aqui são apresentadas as estruturas base que permitem escrever os programas.

Sequências Alternativas

As sequências alternativas utilizam-se quando se pretende que o GRAFCET evolua por uma (ou mais) sequências, de entre as várias disponíveis. Esta sequência tem início com uma divergência OU.

Para ilustrar a divergência OU temos um exemplo representado na Figura 3.6 onde existe a possibilidade de evoluir o Grafcet apenas pela sequência 1 ($a=1$ e $b=0$), apenas pela sequência 2 ($b=1$ e $a=0$) ou por ambas as sequências ($a=1$ e $b=1$). Em cada botão é indicado o valor de a e de b para que não haja qualquer tipo de ambiguidade. Esta estrutura é representada por um traço de ligação horizontal com uma ligação a montante e duas ou mais ligações a jusante.

Na Figura 3.6 temos o exemplo em que é simulada a pressão dos botões a e b que vai activar as duas sequências paralelas (neste caso a alternativa não se verifica). Os sinais gerados pela pressão dos botões podem ser observados no cronograma auxiliar.

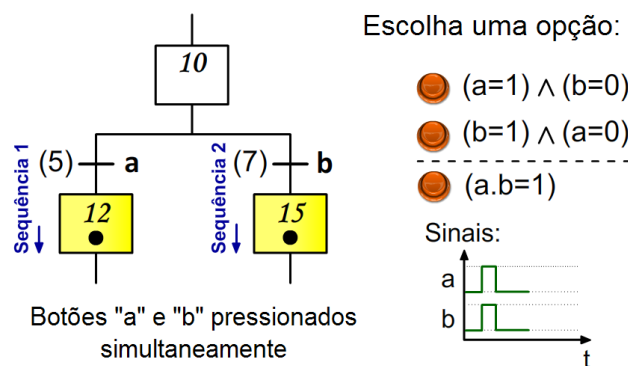


Figura 3.6 – Divergência OU.

No caso da convergência OU a sua representação é um traço de ligação horizontal onde recebe as ligações das sequências a montante e existe apenas uma única ligação a jusante. Para que a transição seja transposta, pelo menos uma das etapas associadas tem de estar activa. Para demonstrar estas situações foram utilizadas duas sequências onde podemos observar a evolução da convergência ao pressionar os respectivos botões. No exemplo da Figura 3.7 o botão b é pressionado ($b=1$ e

$a=0$).

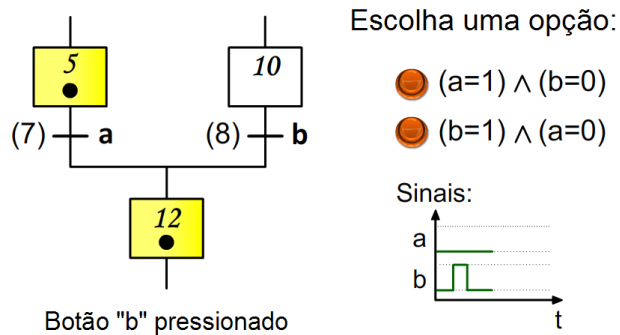


Figura 3.7 – Convergência OU.

Sequências Simultâneas

As sequências simultâneas utilizam-se quando se pretende a activação de várias sequências, a partir de uma determinada etapa. Esta sequência tem início com uma divergência E.

A divergência E é uma situação em que duas ou mais sequências são activadas por uma única transição. Estas sequências vão ser executadas simultaneamente e independentemente umas das outras. Representa-se por um duplo traço horizontal (símbolo de sincronismo) com uma ligação a montante e duas ou mais ligações a jusante.

Para representar a divergência E, o exemplo proposto (Figura 3.8) é constituído por duas sequências que vão ser iniciadas assim que o botão a seja pressionado.

A representação da convergência E é um duplo traço horizontal, conectando as ligações de entrada a uma única ligação de saída. Para que a situação de convergência E possa evoluir é necessário que todas as etapas a montante do símbolo de sincronismo estejam activas e a receptividade da transição de saída seja verdadeira.

Na Figura 3.9 está representada um exemplo de evolução para divergência E. Ao pressionar o botão a , a transição é disparada activando a sequência única seguinte.

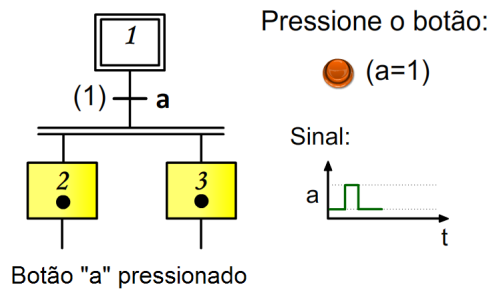


Figura 3.8 – Divergência E.

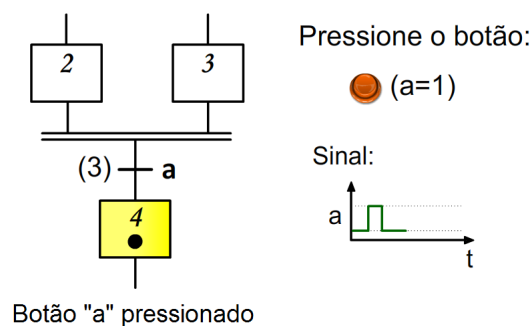


Figura 3.9 – Convergência E.

Escolha de Sequência

A escolha de sequência corresponde a uma divergência OU. Utiliza-se quando se pretende que o Grafcet evolua por uma ou mais sequências, de entre as várias disponíveis, a partir de uma determinada etapa. Quando se pretende que o Grafcet apenas evolua por uma das sequências, as receptividades da divergência OU têm de ser mutuamente exclusivas entre si. Outra maneira de garantir que o Grafcet apenas evolui por uma sequência é estabelecer prioridades entre a evolução das várias sequências, pela alteração apropriada das receptividades associadas às respectivas transições.

No exemplo apresentado na Figura 3.10, são ilustradas as várias situações possíveis pela alteração das funções associadas às receptividades. Quando um botão é pressionado é feita a legenda da opção tomada e os sinais gerados são mostrados no cronograma. Na primeira opção as sequências não têm qualquer tipo de prioridade, existindo as possibilidades de evolução pela sequência 1, pela sequência 2 ou por

ambas. Na segunda opção é dada prioridade à sequência 1, ou seja, no caso dos botões a e b serem pressionados em simultâneo o Grafcet vai evoluir pela sequência 1. Na terceira opção a prioridade é dada à sequência 2, ou seja, no caso de $(a.b=1)$ o Grafcet vai evoluir pela sequência 2. Na quarta opção as receptividades são afectadas com exclusão mútua, aqui garantimos que o Grafcet apenas evolui por uma das sequências (ou nenhuma). Ao pressionar os botões a e b ao mesmo tempo ($a.b=1$) a ordem é ignorada.

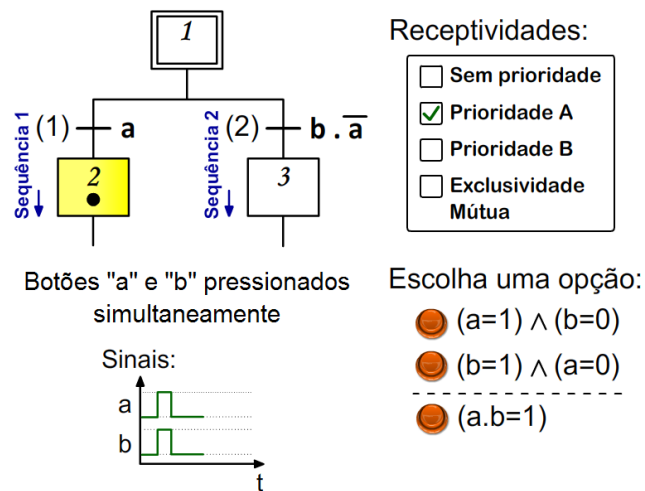


Figura 3.10 – Escolha de sequência 1.

Um outro exemplo de escolha de sequência com receptividades exclusivas é ilustrado na Figura 3.11. Ao pressionar o botão a é executada a sequência 1 e ao pressionar o botão b é executada a sequência 2. Existe um terceiro botão que simula a pressão dos botões a e b em simultâneo. Caso esta situação ocorra as duas sequências são iniciadas em simultâneo. Esta é a forma correcta de utilizar receptividades exclusivas quando se pretende a evolução apenas por uma das sequências, resolvendo-se o problema da necessidade de evolução por ambas as sequências. No exemplo da Figura 3.10 esta situação não é contemplada.

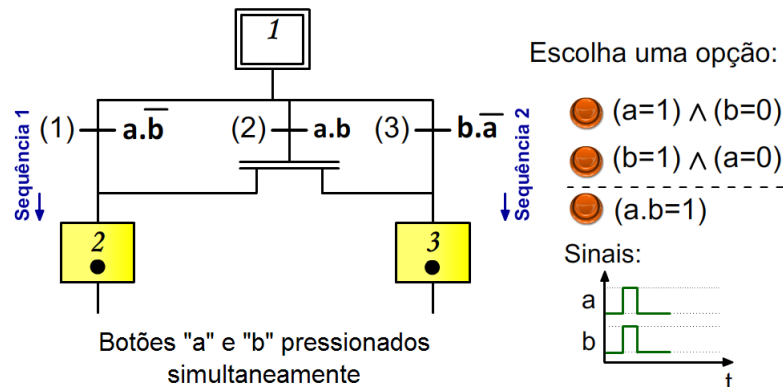


Figura 3.11 – Escolha de sequência 2.

Retoma de Sequência

Quando se pretende repetir um conjunto de etapas (ciclicamente), uma ou mais vezes até que uma determinada condição seja cumprida (por exemplo, contagem ou uma temporização) utiliza-se este tipo de sequência.

No exemplo ilustrado na Figura 3.12 a opção de retomar a sequência ($\bar{a}.p=1$) ou de continuar a evolução do Grafcet ($a.p=1$) é deixada ao critério do utilizador pela escolha do valor lógico a e posterior pressão do botão p . Os sinais gerados por essas opções podem ser visualizados no cronograma auxiliar.

Salto de etapas

O salto de etapas é um caso particular da divergência OU. Permite efectuar um salto de etapas pelo controlo das receptividades. Este tipo de sequência é utilizada quando as acções a realizar por essas etapas não são necessárias para uma determinada condição do automatismo.

O exemplo que ilustra o salto de etapas está representado na Figura 3.13. Neste caso, quando p toma o valor lógico verdadeiro, o valor escolhido para a determina o salto das várias etapas ($a=0$) assim que o botão p é pressionado.

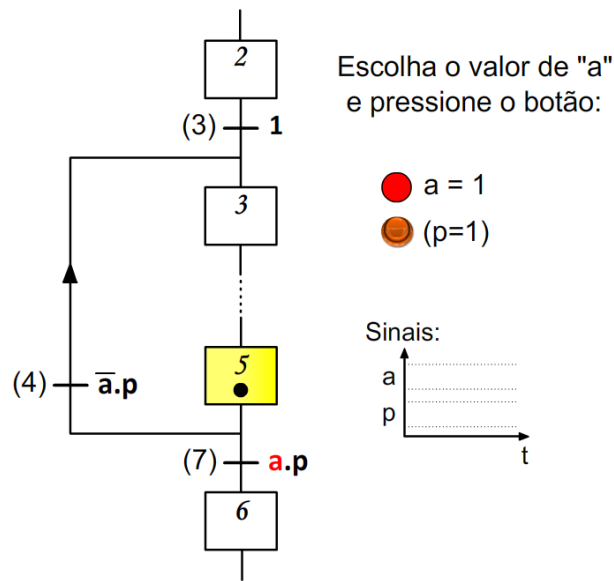


Figura 3.12 – Retoma de sequência.

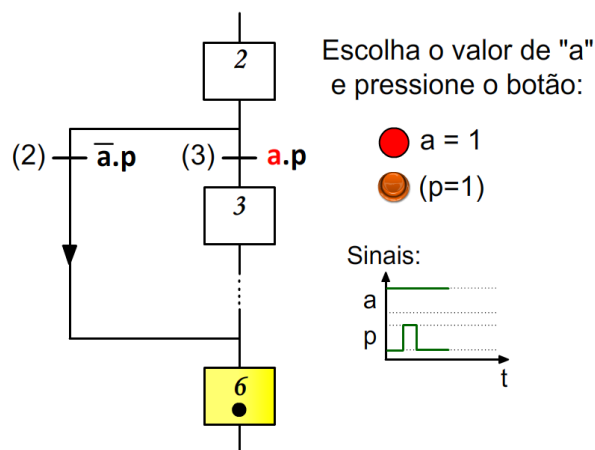


Figura 3.13 – Salto de etapas.

Paralelismo Estrutural

A animação que exemplifica o paralelismo estrutural (ver Figura 3.14) está dividida em várias partes para explicar as diferentes situações de toda a estrutura. É dada a definição de paralelismo estrutural: conjunto de sequências que são activadas de forma simultânea por uma mesma transição. O início desta estrutura é representado por uma transição e um traço duplo horizontal. No final das sequências existe um

traço duplo horizontal seguido de uma única transição. Para que essa transição possa ser disparada todas as etapas a montante terão de estar activas.

Na Figura 3.14, quando a etapa 3 está activa, o disparo da transição 6 ($a=1$) provoca a activação das duas sequências. A evolução das sequências é totalmente independente e no final podem ter que existir "etapas de espera" para fazer o seu sincronismo numa única sequência comum. A transição 18 é disparada no instante em que as etapas 8 e 15 estão activas simultaneamente. Neste exemplo, é associada à transição 18 uma receptividade sempre verdadeira para que o Grafcet possa evoluir rapidamente para a próxima sequência. Não deve ser associada qualquer tipo de acção às etapas de espera.

No final da exposição parcial dos vários conceitos é possível ver a animação completa referente ao paralelismo estrutural e é dada a possibilidade de voltar a rever toda a apresentação.

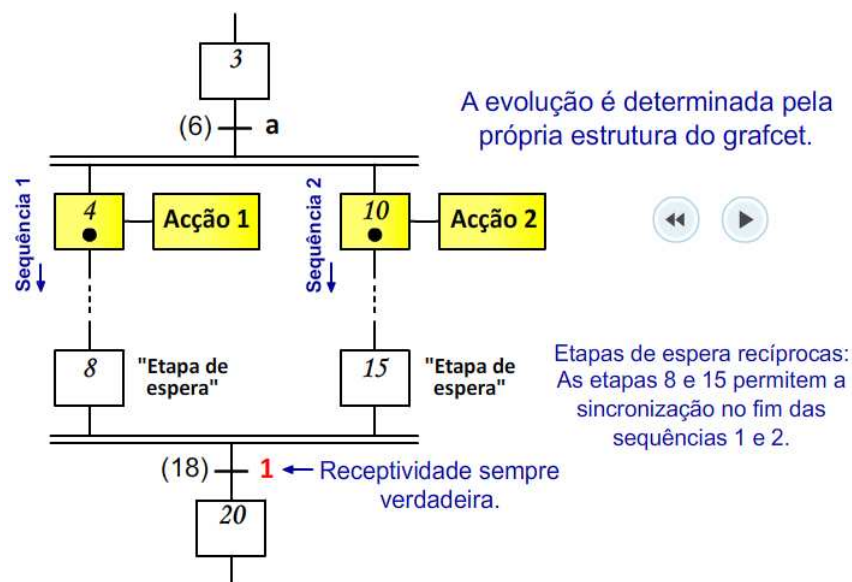


Figura 3.14 – Paralelismo estrutural.

Paralelismo Interpretado

O paralelismo interpretado é iniciado por uma etapa a montante e duas ou mais transições a jusante que vão permitir a escolha de sequência e a sua representação é um simples traço horizontal (ver Figura 3.15). No final de uma selecção de sequência retoma a uma sequência única. O seu nome advém da interpretação das receptividades associadas às transições. A evolução por uma determinada sequência é definida pela receptividade associada à transição no início da sequência. As receptividades devem ser exclusivas quando se pretende a evolução por apenas uma das várias sequências.

Na Figura 3.15 está representada a animação de paralelismo interpretado e pode observar-se a evolução do Grafcet pela sequência 2.

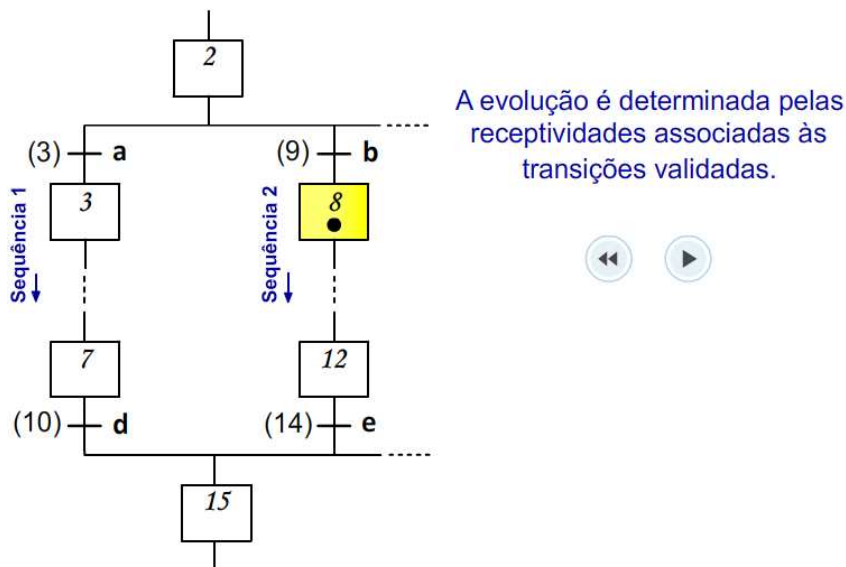


Figura 3.15 – Paralelismo interpretado.

3.3.4 Acções

Na secção seguinte são descritos os vários tipos de acções que podem ser associadas às etapas. As acções dividem-se em dois grandes grupos: **contínuas** e **memorizadas**. Uma ou mais acções podem ser associadas a cada etapa. As acções podem

corresponder a ordens externas para o sistema físico a controlar (por exemplo: ligar um motor, acender uma lâmpada, etc.) ou ordem internas (por exemplo: inicializar um temporizador ou incrementar um contador, etc.).

Acção contínua

Uma acção contínua é executada enquanto a etapa a que está associada estiver activa. Na animação em *flash* ilustrada na Figura 3.16 é feita a descrição e a simulação de uma acção contínua A1. Neste exemplo o disparo da transição 4 ($a=1$) activa a etapa 3, enquanto não ocorrer o disparo da transição 5 ($b=1$) a acção A1 vai ser executada durante esse tempo. Para uma melhor compreensão de todo o funcionamento e das variáveis envolvidas a animação é acompanhada de um cronograma, o qual irá parar nos pontos-chave. No exemplo da Figura 3.16 está indicada a situação 1 (S1) onde ocorre o disparo da transição 4 e a situação 2 (S2) onde ocorre o disparo da transição 5 (activação e desactivação da etapa 3, respectivamente).

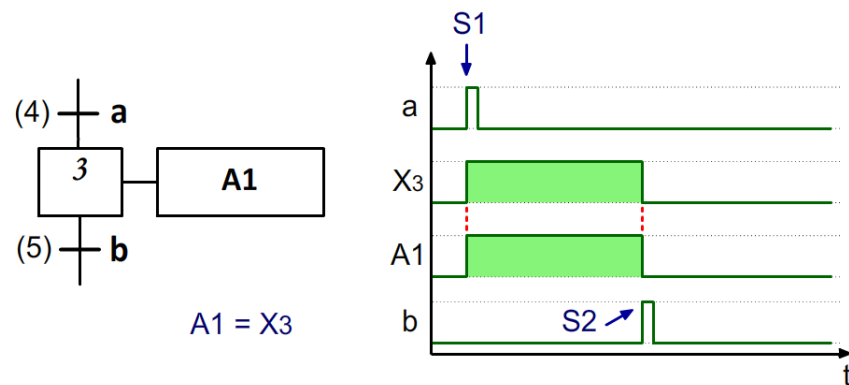


Figura 3.16 – Acção contínua.

Acção condicional

A execução de uma acção condicional está dependente de uma expressão lógica. Essa expressão pode ser uma variável de entrada, uma variável associada a uma etapa ou o resultado Booleano de uma combinação de várias variáveis.

Na animação ilustrada na Figura 3.17 estão descritas várias situações referentes à acção condicional. Neste exemplo, na primeira situação (S1) a condição m é verdadeira mas como a etapa 3 está desactiva a acção A2 não é executada. Na segunda situação (S2) ocorre o disparo da transição 4 ($a=1$) e a etapa 3 é activada, mas como a condição m é falsa a acção A2 não é executada. Na terceira situação (S3) a etapa 3 continua activa e a condição m é verdadeira, logo a acção A2 é executada como se pode observar na Figura 3.17. Na situação 4 (S4), a condição m é falsa e a acção A2 deixa de ser executada. De seguida são simulados mais alguns exemplos e na situação 5 (S5), a acção A2 deixa de ser executada mesmo com a condição m verdadeira, pois o disparo da transição 5 ($b=1$) desactiva a etapa 3 (uma das condições necessárias para que a acção A2 possa ser executada).

No canto inferior esquerdo da Figura 3.17 é apresentada, em paralelo, outra representação equivalente da acção condicional. No final da animação é dada a oportunidade de repetir a animação.

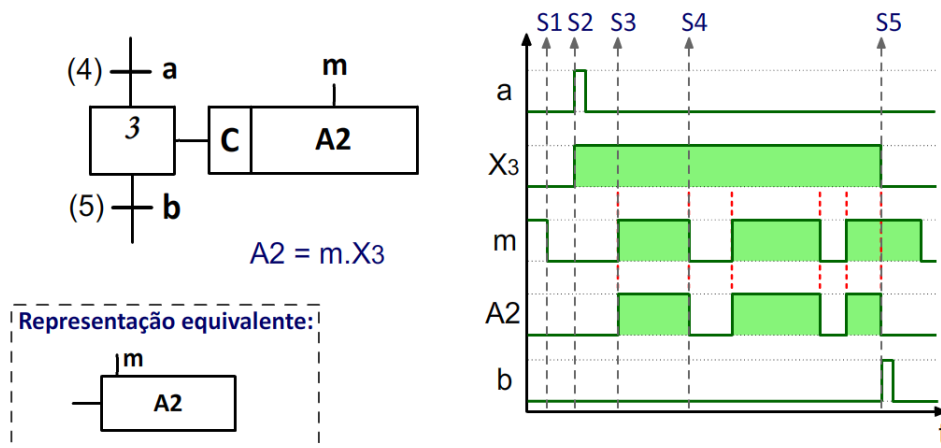


Figura 3.17 – Acção condicional.

Acção limitada

Este tipo de acção significa que a execução da acção está limitada no tempo. A acção é executada durante um tempo T após a activação da etapa associada.

A animação *flash* desta acção (ver Figura 3.18) representa várias situações. De início, na situação 1 (S1), a transição 8 é disparada ($a=1$) o que activa a etapa 10 e a respectiva acção associada A2. O temporizador T é iniciado com um valor de três segundos e a acção 2 é executada. Na situação seguinte (S2) o temporizador T atinge o limite definido e desactiva a execução da acção A2. Para o caso da transição 9 ($b=1$) disparar antes da temporização T terminar, a execução da acção A2 comporta-se como uma acção contínua. No canto inferior esquerdo é apresentada outra representação equivalente da acção limitada. No final a animação em Flash pode ser novamente repetida.

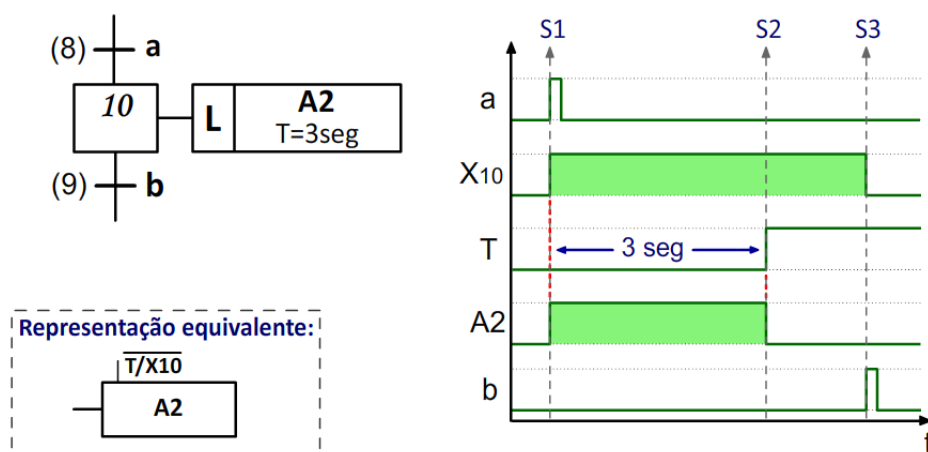


Figura 3.18 – Acção limitada.

Acção retardada

Este tipo de acção retarda a execução da acção. A acção é executada após a activação da etapa associada e decorrido um determinado tempo.

Na animação representada na Figura 3.19 estão ilustradas várias situações. Inicialmente, na situação 1 (S1), é disparada a transição 6 ($a=1$), a etapa 5 é activada e a acção A4 associada não é executada. O temporizador T é iniciado com um valor máximo de três segundos. Na situação seguinte (S2) o temporizador T atinge o limite definido e activa a execução da acção A4. Neste caso, o disparo da transição 7 ($b=1$) deve acontecer passado um tempo superior à temporização T, para garantir que a

acção A4 é executada. Na situação 3 (S3) o disparo da transição 7 ($b=1$) desactiva a etapa 5 e a acção que lhe está associada, A4. No canto inferior esquerdo pode ser visualizada outra representação equivalente da acção retardada, onde também é dada a possibilidade de repetir novamente toda a animação.

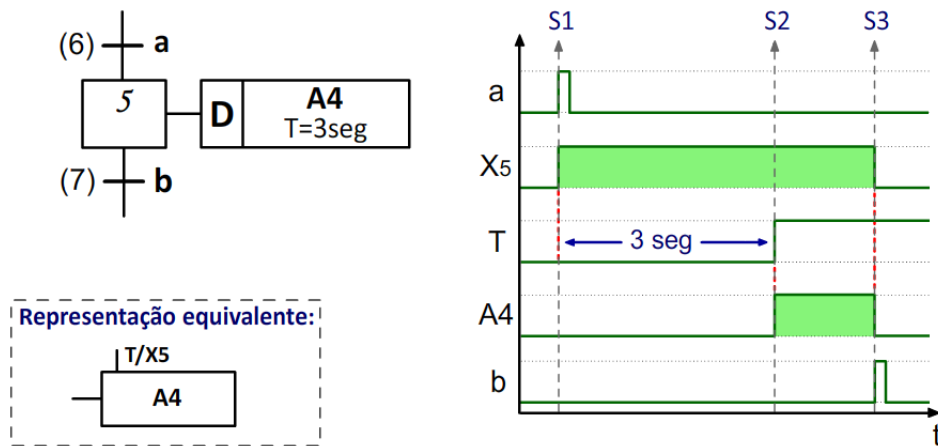


Figura 3.19 – Acção retardada.

Acção memorizada

Este tipo de acção de memorização permite que uma acção fique em execução para além do tempo de activação da etapa associada.

Seguindo o exemplo da animação ilustrada na Figura 3.20, no início (S1) a transição 6 é disparada ($d=1$), a etapa 5 é activada e é feito o *SET* à acção A2, ou seja, vai ser executada até ordem em contrário. Na situação 2 (S2), o disparo da transição 7 ($e=1$) desactiva a etapa 5, mas a acção continua a ser executada. A animação prossegue e na situação 3 (S3), quando a transição 9 é disparada ($h=1$), é feito o *RESET* à acção A2 e essa acção deixa de ser executada (Figura 3.20). Ao mesmo tempo que decorre a animação pode ser observada outra forma de representação equivalente da acção memorizada. A animação pode ser repetida as vezes necessárias.

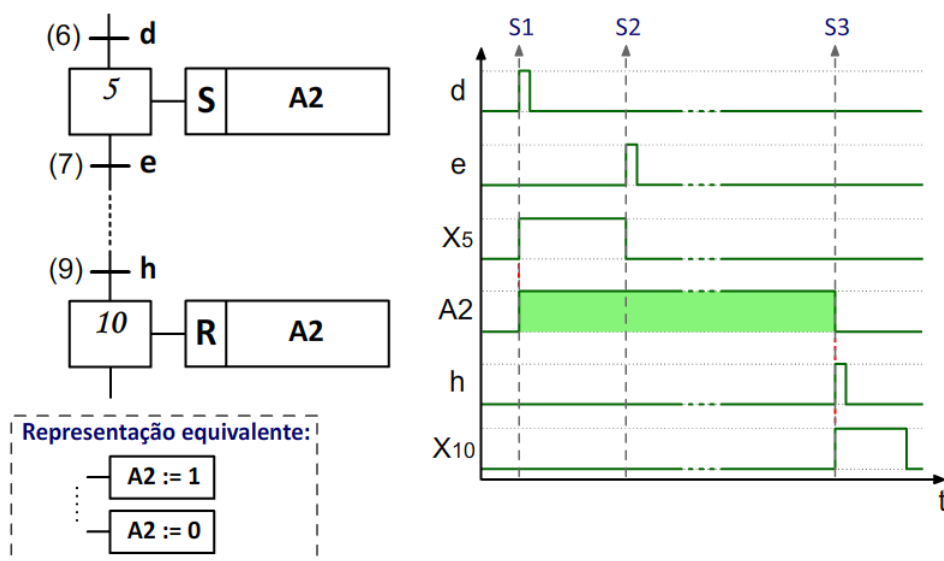


Figura 3.20 – Ação memorizada.

Acção impulsional

A acção impulsional corresponde a uma acção com um tempo de execução muito curto (pulso), mas suficiente para ser reconhecida pelo sistema.

No exemplo apresentado na Figura 3.21, quando ocorre o disparo da transição 4 ($a=1$), a etapa 6 é activada e a acção associada A8 é activada e imediatamente desactivada (activação com a duração de um pulso). Na situação 2 (S2), a etapa 6 é desactivada pelo disparo da transição ($b=1$). Normalmente utiliza-se este tipo de acção para eventos internos, dado que na escala de tempo real a acção não tem tempo suficiente para ser executada - semelhante a uma acção fugaz.

3.3.5 Estruturas Avançadas

Etapa fonte

Uma etapa fonte não tem nenhuma transição a montante. A animação elaborada para ilustrar o princípio de funcionamento da etapa fonte é constituída por três exemplos de activação. O primeiro exemplo de activação refere-se às etapas iniciais que estão activas no início do funcionamento do automatismo. No caso da etapa não

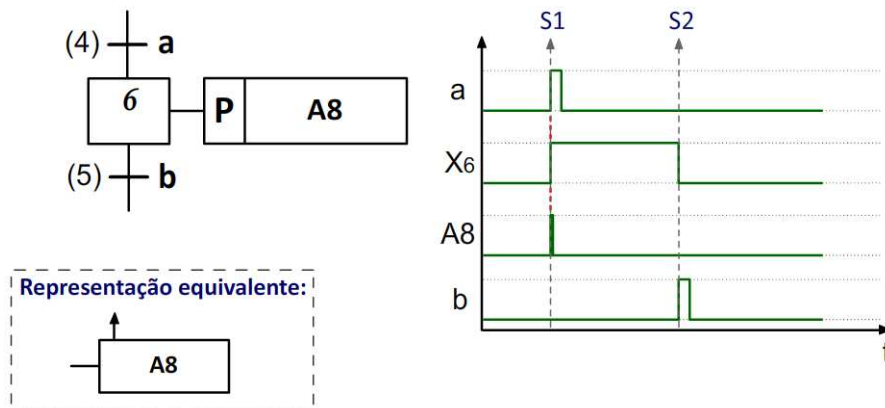


Figura 3.21 – Acção impulsional.

ser a inicial, apenas é activada por uma ordem de forçaçem (a simulação é ilustrada no segundo exemplo de activação) como se pode observar na Figura 3.22, ou por encapsulamento (terceiro exemplo de activação da animação). Notar, no exemplo da Figura 3.22, mesmo que ocorra o disparo da transição 10 ($a=1$) a etapa 8 continua activa enquanto a ordem de forçaçem se verificar. Após a desactivação da etapa 12, o Grafcet G2 evolui livremente.

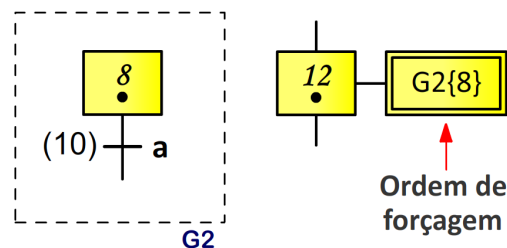


Figura 3.22 – Exemplo de activação forçada de uma etapa.

Etapa poço

Uma etapa poço não tem nenhuma transição a jusante. A animação que ilustra a desactivação de uma etapa poço contém dois exemplos. No primeiro exemplo, a etapa poço é desactivada por uma ordem de forçaçem. No segundo exemplo a desactivação é efectuada por encapsulamento. Entenda-se por encapsulamento um conjunto de etapas (Grafcet parcial) que podem ser encapsuladas por uma

etapa encapsulante. No caso apresentado na Figura 3.23 basta desactivar a etapa encapsulante 15 para desactivar a etapa poço 10 (Figura 3.23), assim como as restantes etapas encapsuladas.

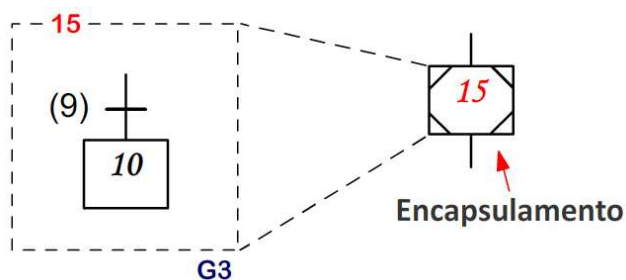


Figura 3.23 – Exemplo de desactivação de uma etapa poço.

Transição fonte

Uma transição fonte não tem nenhuma etapa a montante. A animação ilustrada na Figura 3.24 simula essa transição, que está sempre validada por defeito e por esse motivo geralmente associa-se uma receptividade com flanco ascendente para evitar disparos sucessivos.

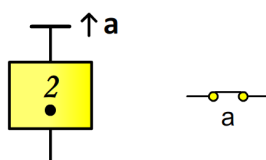


Figura 3.24 – Exemplo de transição fonte.

Transição poço

Uma transição poço não tem nenhuma etapa a jusante. A animação ilustrada na Figura 3.25 simula o disparo da transição poço que tem por objectivo a desactivação da etapa 2. Da mesma forma que a transição fonte, é prudente associar uma receptividade com flanco ascendente.

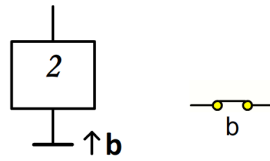


Figura 3.25 – Exemplo de transição poço.

Macro-Etapa

Uma macro-etapa é uma representação de uma sucessão de etapas e transições, o que permite uma melhor representação e compreensão do Grafcet. A simulação ilustrada na Figura 3.26 ilustra o seu funcionamento. A expansão da macro-etapa M5 tem início com a etapa de entrada E5 e termina com a etapa de saída S5, ao longo da sua expansão pode também conter uma ou mais macro-etapas. Notar que, mesmo com a macro-etapa M5 activa e com a receptividade h verdadeira, a transição 10 só é validada e disparada quando a etapa de saída S5 da expansão da macro-etapa M5 é activada.

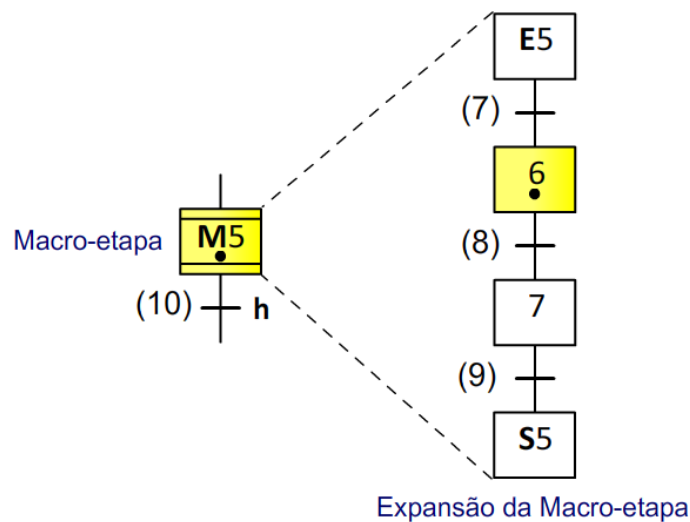


Figura 3.26 – Exemplo de macro-etapa e de macro-expansão.

Sub-programa (tarefa)

Um sub-programa é constituído por um conjunto de várias acções que realizam uma determinada tarefa. No exemplo da animação ilustrada na Figura 3.27, o sub-programa T1 é chamado várias vezes no Grafcet principal. A estrutura do sub-programa é constituída pela etapa de entrada 12 e pela etapa de saída 15, nas quais não deve ser associada nenhuma acção exterior. Na animação são observados vários pontos-chave. No exemplo da Figura 3.27 é ilustrado o disparo da transição 9 pela activação da etapa 1. Para que isto aconteça, na função de receptividade associada à transição 9 devem constar todas as etapas onde a tarefa é chamada, para que esta possa ser iniciada. A partir deste momento o sub-programa evolui normalmente. A função de receptividade da transição a montante da etapa onde a tarefa é chamada (transição 2, por exemplo) deve ser associada à etapa de saída do sub-programa, para quando este terminar o Grafcet principal possa evoluir normalmente. Assim que a transição 2 é disparada, a etapa 1 é desactivada e o sub-programa volta ao seu estado inicial. Este ciclo recomeça novamente quando a etapa 3 é activada.

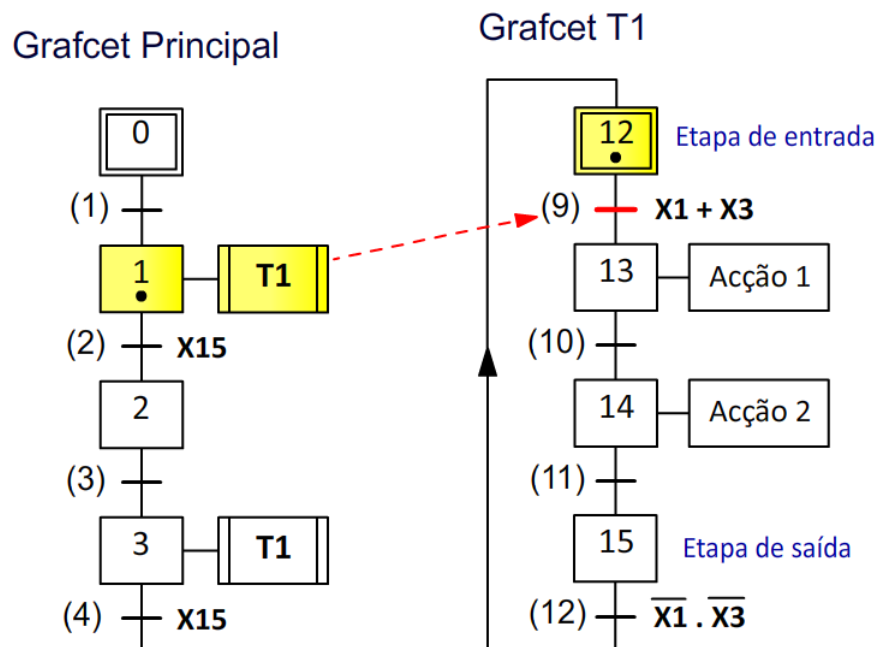


Figura 3.27 – Exemplo de sub-programa (T1).

Receptividade dependente do tempo

Uma receptividade pode estar dependente do tempo (ver Figura 3.28). A notação utilizada é $t1/d/t2$ em que a receptividade associada à transição 2 é verdadeira após um tempo $t1$ (depois de ocorrer um flanco ascendente da variável d). Volta novamente a ser falsa após um tempo $t2$ (depois de ocorrer um flanco descendente da variável d). Quando é feita uma temporização a variável d deve permanecer activa durante um tempo superior a $t1$ para que a receptividade seja verdadeira. Primeira situação (S1): a etapa 3 está activa e valida a transição 2. A transição ascendente da variável d inicia a temporização $t1$ de 2 segundos. Segunda situação (S2): é ilustrado um exemplo onde o tempo de activação da variável d não foi suficiente para terminar a temporização, logo a transição não é disparada. Na terceira situação (S3), ocorre novamente uma transição ascendente da variável d que inicia novamente a temporização. Quarta situação (S4): a temporização $t1$ atinge o valor definido de 2 segundos, a receptividade, $r2$, é verdadeira e ocorre o disparo da transição (etapa 4 activa). Após isto a condição de receptividade continua verdadeira, ou seja, caso a etapa 3 seja activada ocorre novamente o disparo da transição. Quinta situação (S5): transição descendente da variável da variável d , é iniciada a temporização $t2$ (1 segundo). A condição de receptividade continua verdadeira durante esse tempo. Na situação seguinte (S6), a temporização $t2$ atinge o valor definido de 1 segundo e a condição da receptividade é falsa. Todos os sinais envolvidos são ilustrados no cronograma para melhor compreensão das várias situações. Na Figura 3.28 pode-se observar a representação das temporizações $t1$ e $t2$ respectivamente.

3.4 Exemplos

Nesta secção são apresentados os exemplos de modelação de controladores lógicos em Grafcet e respectiva animação. O controlo da animação é simulado com a implementação do programa em Grafcet num PLC. Cada exemplo segue a seguinte forma: de início é explicado o enunciado do problema e a estrutura que se pretende demonstrar, de seguida são apresentadas as resoluções propostas para cada alínea desse exemplo e são abordados os seus pontos essenciais.

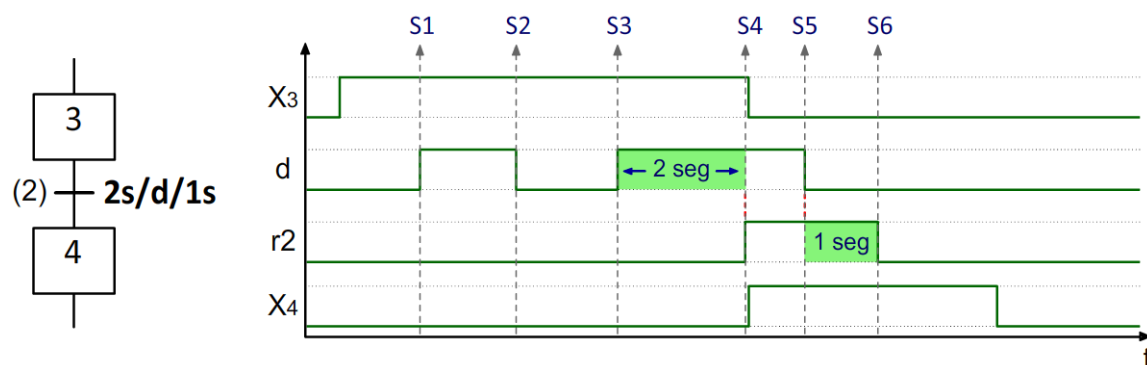


Figura 3.28 – Exemplo de receptividade dependente do tempo.

3.4.1 Exemplo 1: Ciclo de uma só sequência

O exemplo 1 [10] é referente ao *ciclo de uma só sequência*. São propostos quatro problemas para a sua implementação em Grafcet.

Enunciado do problema:

Considera-se um carro que se pode movimentar entre as posições A e B, como se ilustra na Figura 3.29.



Figura 3.29 – Exemplo 1.

A presença do carro em A é representada pela variável Booleana $a=1$ e em B é representada por $b=1$. O carro é comandado por um botão m ($m=1$ quando pressionado). Inicialmente o carro está na posição A (repouso). As saídas do sistema de controlo são D e E (D=1, o carro desloca-se para a direita; E=1, o carro desloca-se para a esquerda). Com a utilização do *GRAF CET* vão ser modelados os controladores lógicos que correspondem às situações dos exemplos seguintes.

Exemplo 1.1

Enunciado: Quando a variável m transita do valor lógico '0' para o valor lógico '1' (apenas quando o botão é pressionado) o carro inicia o movimento e faz o percurso ABA. Qualquer que seja o valor do botão m quando o carro chega a A, este pára e espera pela próxima mudança da variável m de '0' para '1'.

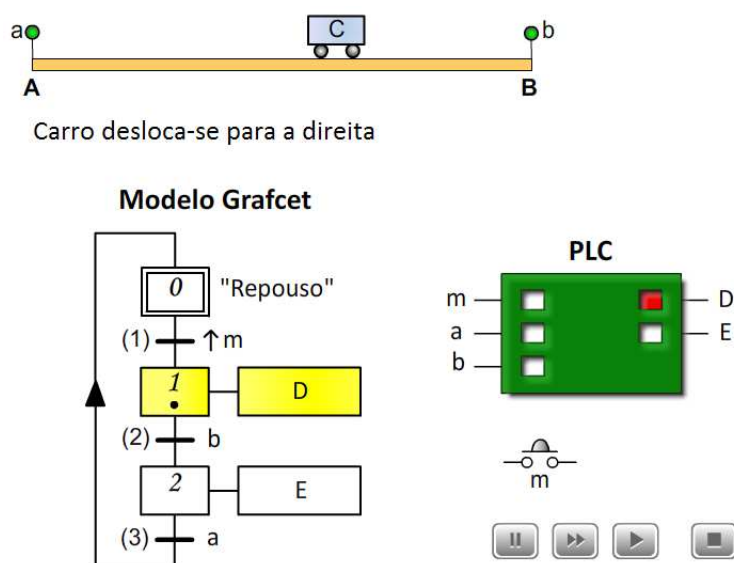


Figura 3.30 – Ciclo de uma só sequência - Exemplo 1.1.

Para a implementação física do problema temos um carro que pode deslocar-se para a direita ($D=1$) e para a esquerda ($E=1$), dois sensores de fim de curso, a e b , que detectam a presença do carro nesses pontos e um botão m .

O que se pretende transmitir com este exemplo é a implementação do ciclo de uma só sequência (Figura 3.30). A transição 1 está associada a uma receptividade com flanco ascendente. Esta condição permite iniciar a sequência apenas quando o botão m é pressionado e impede o carro de começar novo ciclo caso o botão esteja pressionado no momento do regresso à posição inicial. O carro só vai iniciar nova viagem quando ocorrer uma transição ascendente da variável m . As restantes transições estão associadas a variáveis Booleanas que representam os sensores de proximidade.

As acções associadas às etapas são contínuas e vão ser executadas enquanto a respectiva etapa estiver activa.

Exemplo 1.2

Enunciado: Neste exemplo, ilustrado na Figura 3.31, o botão m é substituído por um interruptor. Quando o interruptor é accionado (variável m passa do valor '0' para o valor '1') o carro inicia o movimento e faz o percurso ABA. Independentemente do valor de m quando o carro chega a A, este pára e espera pela próxima transição ascendente (de '0' para '1') da variável m . Uma lâmpada V é accionada quando o carro está em repouso.

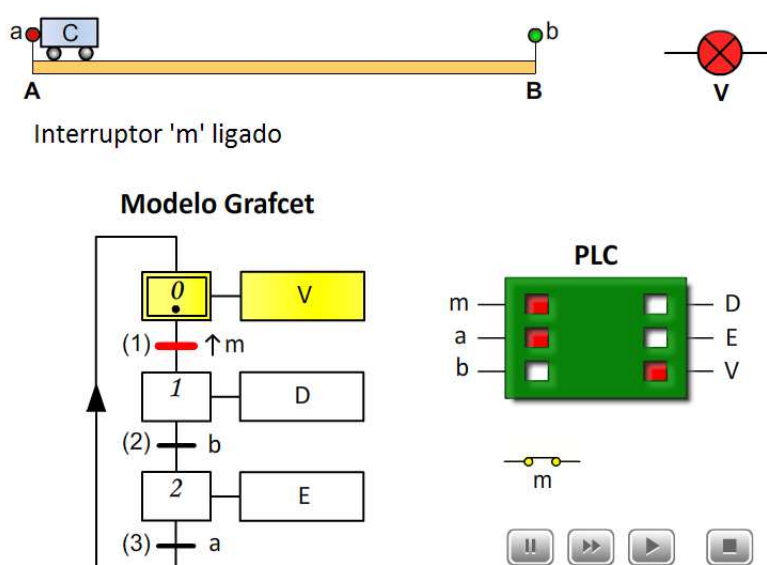


Figura 3.31 – Ciclo de uma só sequência - Exemplo 1.2.

Como pontos essenciais temos a transição ascendente da variável associada ao interruptor m que inicia a sequência. O uso do interruptor permite verificar que a sequência apenas volta a ser executada quando este transita de '0' para '1' lógico, ou seja, é necessário abrir e voltar a fechar o interruptor.

Exemplo 1.3

Enunciado: Este exemplo está ilustrado na Figura 3.32. Quando o interruptor é pressionado o carro inicia o movimento e faz o percurso ABA. Se $m=1$ quando o carro chega ao ponto A, este continua o movimento para um novo ciclo ABA, caso contrário pára. Uma lâmpada V é accionada quando o carro está em repouso.

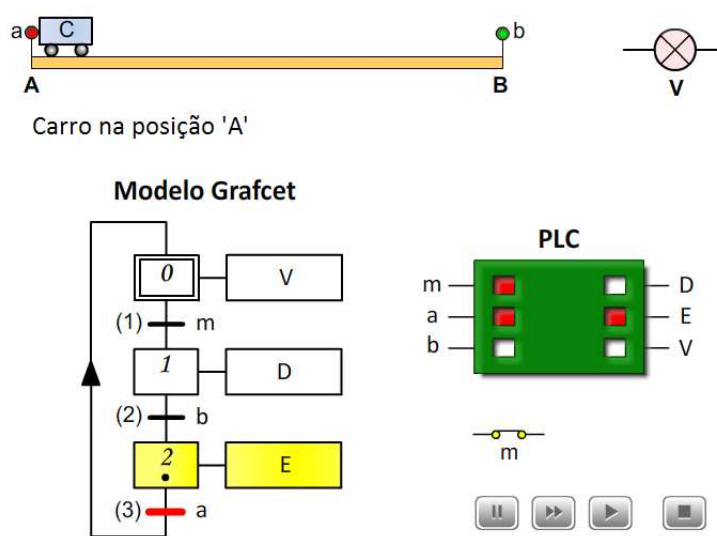


Figura 3.32 – Ciclo de uma só sequência - Exemplo 1.3.

A sequência é iniciada quando se liga o interruptor m . A condição de transição da variável m foi retirada, logo enquanto o interruptor estiver ligado a sequência está sempre a ser executada.

É introduzida a noção de *acção fugaz* onde as *acções contínuas* não são executadas devido à rápida evolução do Grafcet. O mesmo não ocorre para as *acções memorizadas*. Na situação deste exemplo, as transições 3 e 1 vão ser disparadas sucessivamente (devido a $m=1$) o que vai impedir a acção associada à etapa 0 (accionar a lâmpada V) de ser executada (Figura 3.32). O ciclo vai ser repetido indefinidamente até desligar o interruptor ($m=0$).

Exemplo 1.4

Enunciado: É accionada uma lâmpada V enquanto o carro está na posição de repouso. Ao carregar no interruptor o carro inicia o movimento e faz o percurso ABA. Quanto atinge o ponto B o carro vai aguardar 1 segundo, antes de inverter o sentido do movimento (ver Figura 3.33). Ao chegar ao ponto A se $m=1$ o carro continua o movimento para um novo ciclo ABA, caso contrário pára. O número de viagens que o carro pode efectuar está limitado a 3.

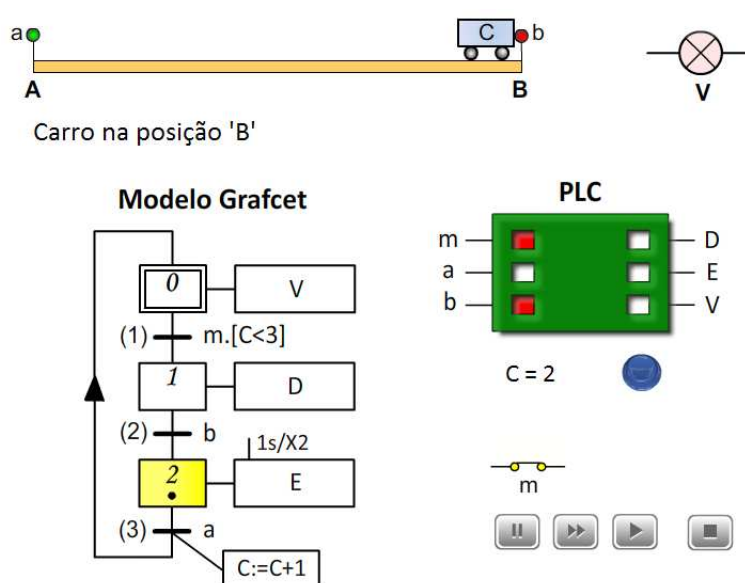


Figura 3.33 – Ciclo de uma só sequência - Exemplo 1.4.

A sequência é iniciada ao ligar o interruptor m . Como no exemplo anterior, enquanto o interruptor m estiver ligado a sequência está sempre a ser executada. Para terminar este ciclo é necessário desligar o interruptor.

A noção de *acção fugaz* está presente neste exemplo. Quando o carro está no ponto B temos o exemplo de uma acção retardada (etapa 2). A acção de movimento do carro para a esquerda vai ser executada 1 segundo após a activação da etapa correspondente.

As viagens efectuadas pelo carro são contabilizadas por um contador C. Este contador é incrementado na transição 3 - *acção na transição*. Aqui é dada a noção do uso de contadores e o exemplo de uma acção que pode ser associada às transições. O contador ao atingir o valor 3 (máximo) mantém o carro na posição inicial. Isto acontece pois a condição do *predicado* $[C < 3]$ associada à receptividade da transição 1 é falsa. Nesta situação o Grafcet fica bloqueado, para poder continuar a animação e visualizar as várias situações que esta permite, existe um botão auxiliar de cor azul em que a sua função é inicializar o contador a 0. Isto mesmo poderia ser implementado no Grafcet.

3.4.2 Exemplo 2: Paralelismo Interpretado

O exemplo 2 é referente ao *paralelismo interpretado*. São propostos dois problemas para a sua implementação em GRAFCET, um com a aplicação de exclusão mútua e outro com prioridades.

Enunciado do problema:

Considere-se o seguinte sistema de transporte de material para dois locais B e C representado na Figura 3.34.

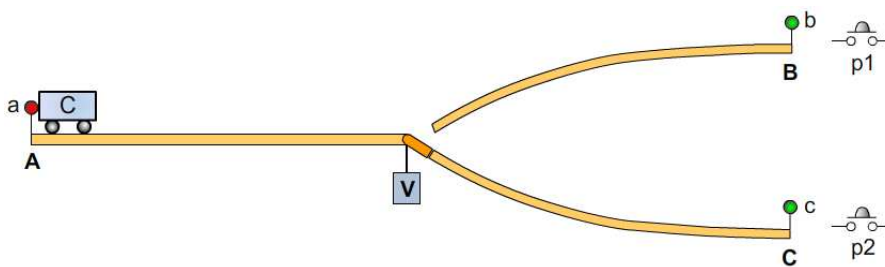


Figura 3.34 – Exemplo 2.

Nos pontos A, B e C existem sensores de fim de curso: a, b e c . Inicialmente o carro está na posição A (repouso). O carro pode ser comandado por dois botões $p1$ e $p2$. Se o botão $p1$ for pressionado o carro efectua o trajecto ABA. Caso seja pressionado o botão $p2$ o carro efectua o trajecto ACA. Os motores D e E movimentam o carro para a direita e para a esquerda, respectivamente. O actuador V controla uma

agulha da seguinte forma: $V=0$ - selecciona a direcção C (posição de repouso da agulha); $V=1$ - selecciona a direcção B. Com a utilização do *GRAFCET* vão ser modelados os controladores lógicos que correspondem às situações seguintes.

Exemplo 2.1

Enunciado: Considere-se o Grafcet apresentado na Figura 3.35. Ao premir o botão $p1$ o Grafcet evolui por uma sequência (percurso ABA) e ao premir o botão $p2$ o Grafcet evolui por uma sequência diferente (percurso ACA). Em qualquer dos casos o carro volta à posição inicial e aguarda nova ordem. Esta ordem é ignorada quando os botões $p1$ e $p2$ são pressionados ao mesmo tempo.

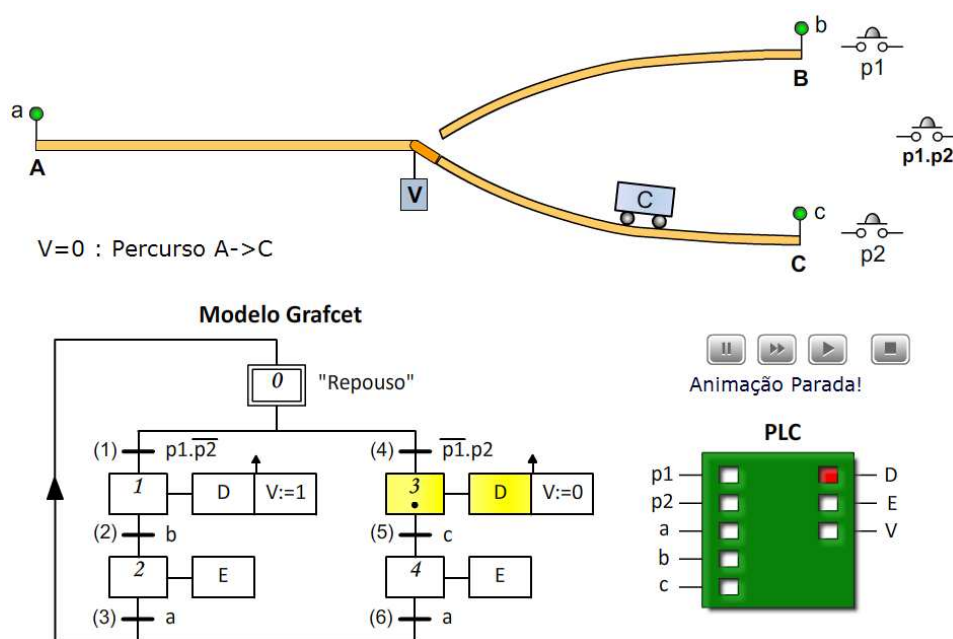


Figura 3.35 – Paralelismo interpretado - Exemplo 2.1.

Este exemplo permite verificar a aplicação de paralelismo interpretado. As receptividades associadas às várias transições no início do paralelismo vão ser "interpretadas" de modo a decidir qual ou quais as sequências a activar.

Neste *Grafcet* (Figura 3.35), as receptividades relativas à escolha da sequência são implementadas com exclusão mútua. Sem a exclusão mútua, no caso em que os

botões $p1$ e $p2$ são pressionados em simultâneo o Grafcet iria evoluir para uma situação instável. O botão auxiliar $p1.p2$ permite simular uma ordem simultânea. Com o uso de exclusão mútua esta ordem é ignorada e mantêm-se a estabilidade do sistema. Assim, para que o Grafcet evolua, apenas pode ocorrer uma das ordens ($p1$ ou $p2$).

A *acção na activação* está presente neste exemplo pela escolha de direcção no actuador V. A escolha de direcção é realizada pela agulha V dependendo do percurso escolhido. A ordem é transmitida e memorizada na activação da etapa 1 (percurso ABA) ou etapa 3 (percurso ACA) - *acção na activação*.

Exemplo 2.2

Enunciado: Este exemplo, representado pela Figura 3.36, difere do anterior na escolha da sequência. Se os botões $p1$ e $p2$ forem pressionados simultaneamente o carro vai executar o trajecto ABA (prioridade para botão $p1$), como se pode observar na Figura 3.36.

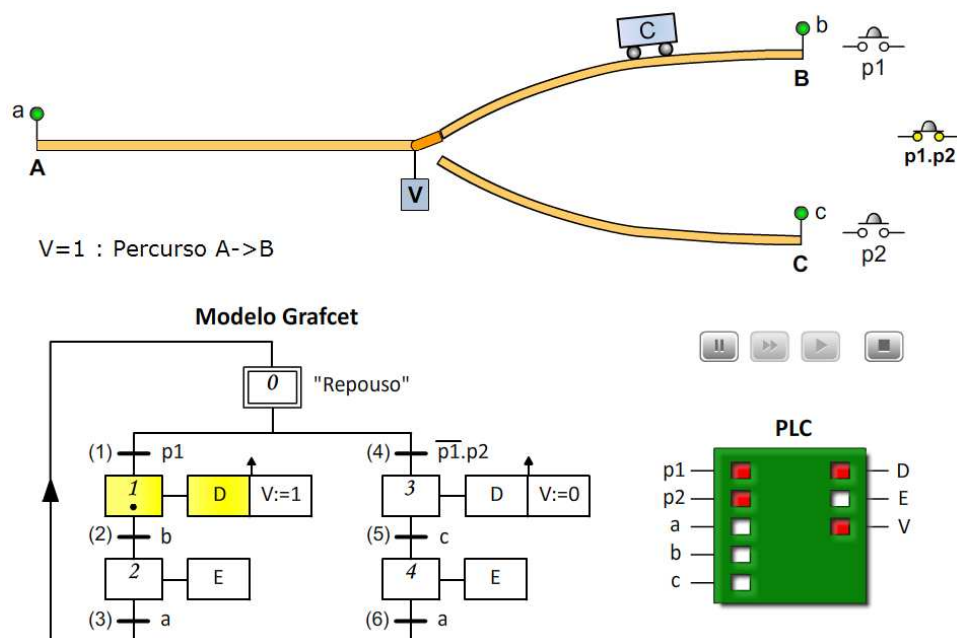


Figura 3.36 – Paralelismo interpretado - Exemplo 2.2.

Neste caso, é apresentada outra solução que passa pela aplicação de prioridades. Na situação em que os botões $p1$ e $p2$ são pressionados em simultâneo é executada pelo menos uma das sequências - prioridade para o botão $p1$ (percurso ABA).

3.4.3 Exemplo 3: Paralelismo Estrutural

O exemplo 3 é referente ao *paralelismo estrutural*. São propostos três problemas para a sua implementação em GRAFCET.

Enunciado do problema:

Considere-se dois carros como estão representados na Figura 3.37.

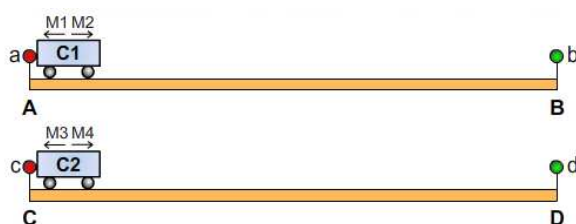


Figura 3.37 – Exemplo 3.

Na situação de repouso os carros encontram-se no extremo esquerdo do percurso. Os motores M2 e M4 deslocam para a direita os carros C1 e C2, respectivamente. Os motores M1 e M3 movem os carros C1 e C2 para a esquerda. Nos pontos A, B, C e D existem os sensores fim de curso: a , b , c e d , que detectam a presença dos carros. Ao pressionar o botão p ambos os carros devem efectuar uma viagem de ida e volta (ABA e CDC). Os carros deslocam-se de forma independente um do outro, pois mediante a carga que transportam podem movimentar-se a velocidades diferentes. Com a utilização do *GRAFCET* vão ser modelados os controladores lógicos que correspondem às situações seguintes.

Exemplo 3.1

Enunciado: Considere-se o Grafcet representado na Figura 3.38. Ao premir o botão p é iniciado o movimento dos carros C1 e C2. No regresso, apenas é permitida

uma nova viagem quando ambos os carros se encontram na situação de repouso. O carro que se deslocar mais rápido vai esperar pelo carro mais lento, antes de poder iniciar novo ciclo.

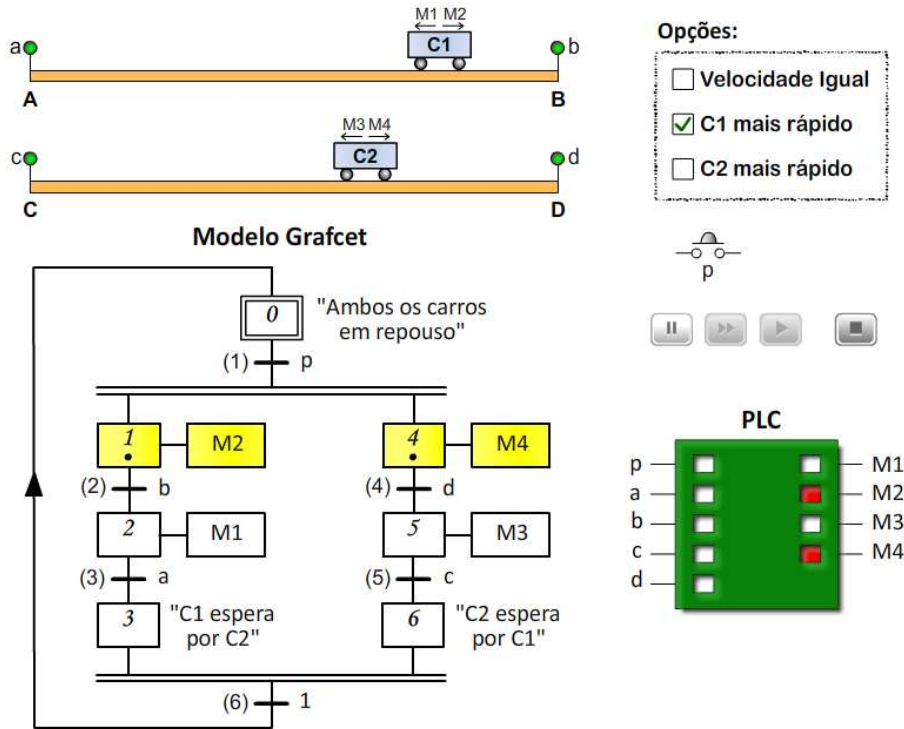


Figura 3.38 – Paralelismo estrutural - Exemplo 3.1.

Este exemplo permite compreender o uso e aplicação do paralelismo estrutural. O início de evolução das várias sequências é definido pela estrutura do paralelismo. Ao pressionar o botão *p* são iniciadas as duas sequências em paralelo, (Figura 3.38). Cada sequência evolui de forma independente, sendo esta evolução mais rápida ou mais lenta dependendo da rapidez dos carros (carga a transportar). A carga, ou melhor, a velocidade de deslocamento dos carros fica ao critério do aluno. Para o efeito, o exemplo dispõe de um menu de opções onde podem definir 3 tipos de velocidades, de modo a simular o peso da carga a transportar: mesma carga (velocidade igual), mais carga para o carro 2 (carro 1 mais rápido) ou mais carga para o carro 1 (carro 2 mais rápido). A etapa de espera no final das sequências permite fazer o sincronismo do movimento dos carros. A receptividade associada à

transição 6 está sempre verdadeira. Isto permite o seu disparo imediato no instante em que as etapas 3 e 6 forem activas simultaneamente.

Exemplo 3.2

Enunciado: Este exemplo aborda o mesmo problema apresentado no exemplo 3.1, apresentando uma solução que utiliza *GRAFCETs parciais*.

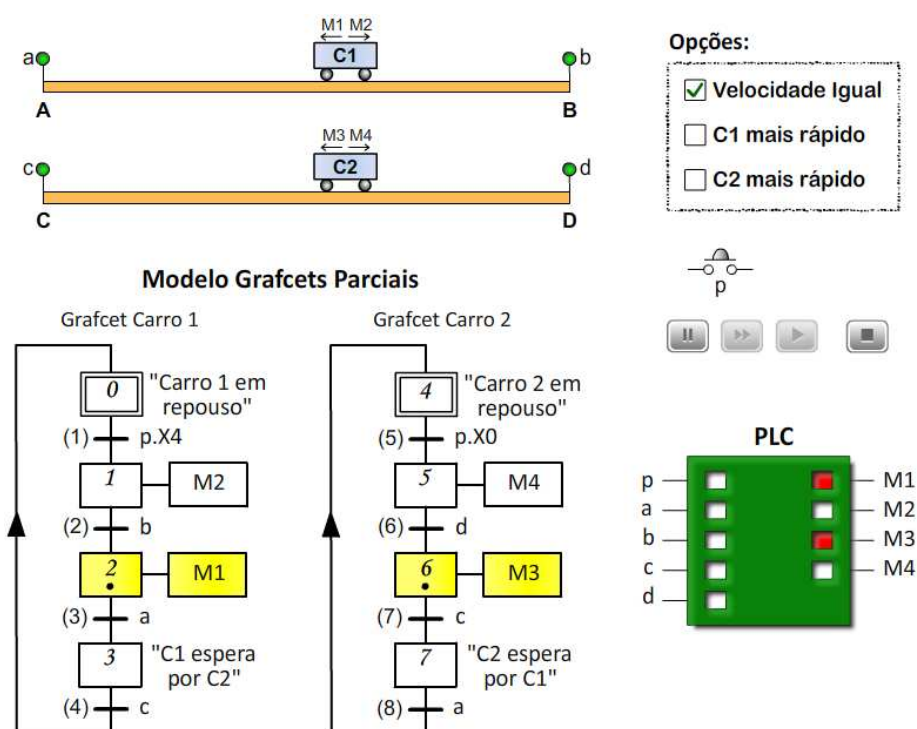


Figura 3.39 – Paralelismo estrutural - Exemplo 3.2

Neste exemplo, representado na Figura 3.39, cada Grafcet está adaptado ao respectivo elemento activo da animação (carro). O sincronismo inicial dos carros é assegurado pelas condições das receptividades das transições 1 e 5. Após pressionar o botão *p*, o carro 1 avança se o carro 2 estiver também em repouso. O mesmo se verifica para o Grafcet parcial do carro 2. Desta forma fica garantido que os carros iniciam o movimento ao mesmo tempo. As etapas de espera são mantidas e a receptividade que estava sempre validada (transição 6 do exemplo anterior) é substituída pela variável associada ao sensor de movimento do carro oposto (quando

paralelas, quando os carros C1 e C2 se encontram na posição B e D respectivamente. Para o efeito são introduzidas mais duas etapas de espera.

3.4.4 Exemplo 4: Recurso Partilhado

O exemplo 4 é referente à utilização de um *recurso partilhado*. São propostos três problemas para a sua implementação em GRAFCET.

Enunciado do problema:

Vamos considerar dois carros de transporte que servem dois postos de carga, como se pode observar na Figura 3.41.

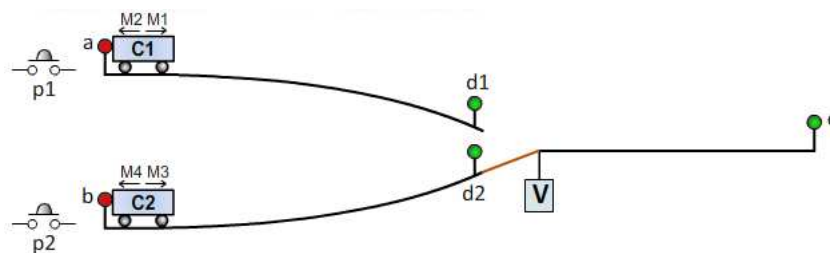


Figura 3.41 – Exemplo 4.

Inicialmente os carros encontram-se na situação de repouso. Ao pressionar o botão $p1$ o carro 1 efectua o percurso ACA. Se o botão $p2$ for pressionado o carro 2 efectua o percurso BCB. Quando os carros retomam à posição inicial (A e B) permanecem em repouso aguardando nova ordem. Os pedidos são ignorados quando os carros não se encontram na posição de repouso. O motor M1 desloca o carro 1 para a direita, o mesmo acontece com o motor M3 para o carro 2. O motor M2 desloca o carro 1 para a esquerda, o motor M4 tem o mesmo comportamento no carro 2. A direcção da agulha é controlada pela electroválvula V. Quando $V=0$, a agulha orienta-se para o percurso BCB; quando $V=1$ o percurso é ACA. Um carro ao chegar à zona de agulhagem D, só poderá seguir em direcção a C caso o percurso D-C esteja livre. Utilizando o GRAFCET vão ser modelados os controladores lógicos que correspondem aos seguintes exemplos.

Exemplo 4.1

Enunciado: Ao premir o botão $p1$ ou o botão $p2$, o respectivo carro inicia o movimento em direcção ao ponto C. Um carro ao chegar à zona de agulhagem verifica se o percurso partilhado está livre. Se está avança, caso contrário espera até que este fique livre. Na situação dos carros chegarem à zona de agulhagem D ao mesmo tempo, é dada a prioridade de avançar ao carro 1.

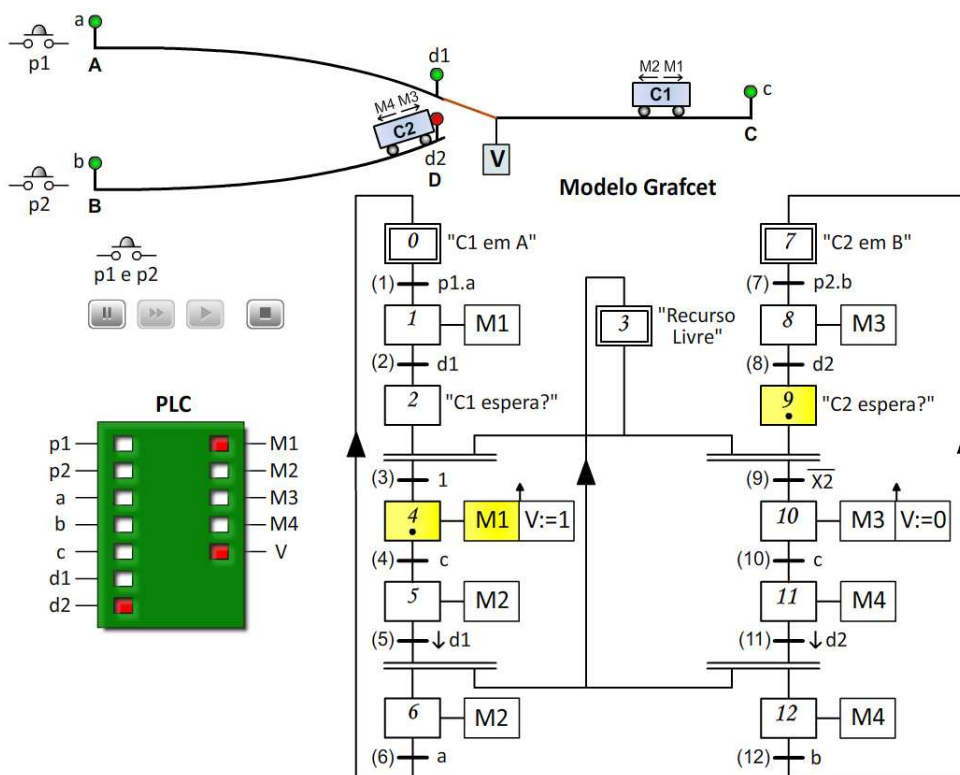


Figura 3.42 – Recurso partilhado - Exemplo 4.1.

Neste exemplo é necessário controlar o acesso a uma zona do percurso que é partilhada por dois carros. Para o efeito (ver Figura 3.42), é utilizado um semáforo (etapa 3) para sinalizar o recurso livre e duas convergências E para permitir o acesso dos carros à zona partilhada.

A receptividade associada à transição 3 que dá a ordem para o carro 1 avançar é sempre verdadeira e a do carro 2 é $\overline{X2}$. Desta forma atribui-se prioridade ao carro

1. O carro 2 avança quando o recurso partilhado está livre e o carro 1 não está à espera em D. As acções associadas aos motores são contínuas e a acção na activação define o valor da variável de controlo da electroválvula V. As transições 5 e 11 têm associadas variáveis com flanco descendente. Quando uma delas é verdadeira indica que a zona de agulhagem acabou de ficar livre e o semáforo (etapa 3) é activado.

Exemplo 4.2

Enunciado: Neste exemplo, ilustrado na Figura 3.43, o controlo dos carros é modelado pelo uso de *Grafkets* parciais.

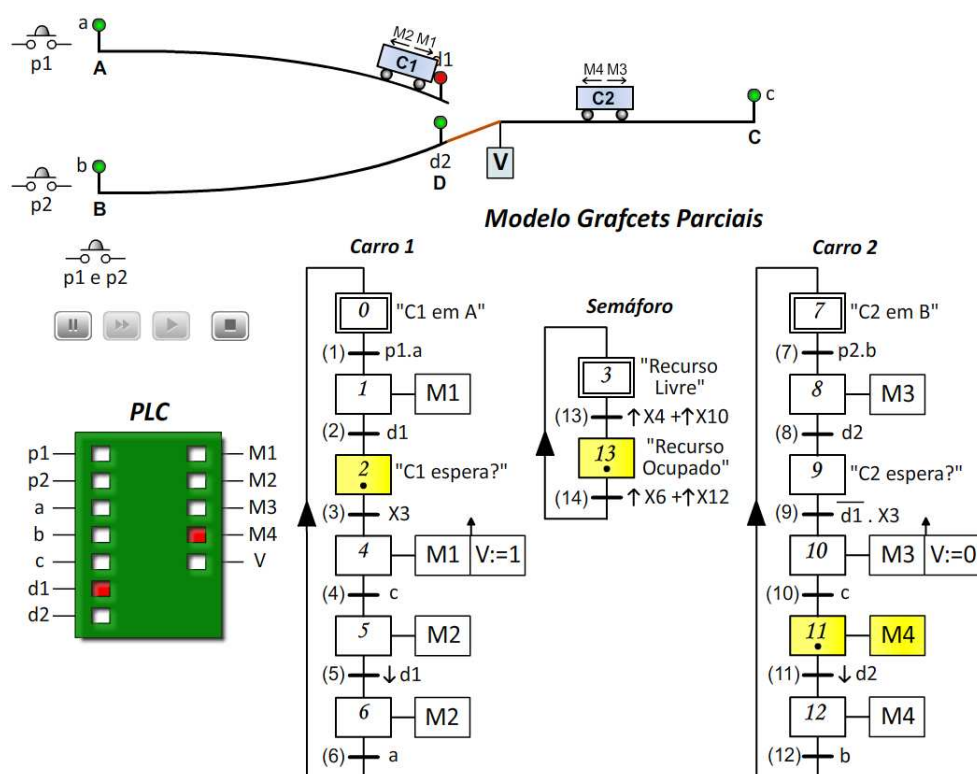


Figura 3.43 – Recurso partilhado - Exemplo 4.2.

É apresentada uma solução que permite visualizar o controlo das partes activas (carros e agulha) de forma independente, como se pode observar na Figura 3.43. O Grafket parcial relativo ao semáforo é constituído por duas etapas que indicam que o recurso está livre ou está ocupado. A condição da transição 13 refere-se às

etapas onde os carros iniciam a entrada na zona de agulhagem (X4 para o carro 1 e X10 para carro 2). A condição da transição 14 é composta pelas etapas dos carros referentes à saída da zona de agulhagem (X6 para o carro 1 e X12 para o carro 2).

A condição de receptividade associada à transição 3, no exemplo anterior, que está sempre validada é substituída pela condição X3 (indicação de recurso livre). A condição de receptividade, no exemplo anterior, associada à transição 9 é adaptada de $\overline{X2}$ para $\overline{d1.X3}$, isto é, o carro 2 avança quando o carro 1 não está em $d1$ (início da zona de agulhagem) e o recurso partilhado está livre. Desta forma é mantida a prioridade para o carro 1.

Exemplo 4.3

Enunciado: O exemplo (Figura 3.44) é o mesmo que se apresentou em 4.2. Aqui os carros vão ter acesso ao percurso $D-C$ alternadamente. Inicialmente apenas o carro 1 poderá efectuar o percurso em direcção a C. O próximo carro a poder fazer esse percurso é o carro 2. Se o carro 1 tentar efectuar novamente o percurso ACA, antes do carro 2, vai ter de esperar na zona de agulhagem até que o carro 2 utilize o percurso $d2-C$. E assim sucessivamente. . . Um carro nunca poderá utilizar o percurso partilhado duas vezes consecutivas.

A solução para este exemplo (Figura 3.44) permite perceber uma forma de implementar regras de acesso a recursos partilhados. Tudo o que foi dito para o exemplo 4.1 é válido para este, exceptuando a parte que concerne à receptividade da transição 9. Neste caso é idêntica à receptividade da transição 3, assim que o carro recebe a indicação que a zona de agulhagem está livre então pode avançar para o recurso partilhado.

A regra de acesso que permite a utilização do recurso partilhado de forma alternada é aplicada com a utilização de um segundo semáforo (etapa 13), Figura 3.44. Um semáforo nunca está activo duas vezes consecutivas, isso garante o acesso alternado dos carros ao recurso partilhado. O semáforo activo inicialmente é a etapa 3 dando prioridade de acesso ao carro 1. Assim que o carro 1 libertar o recurso partilhado vai

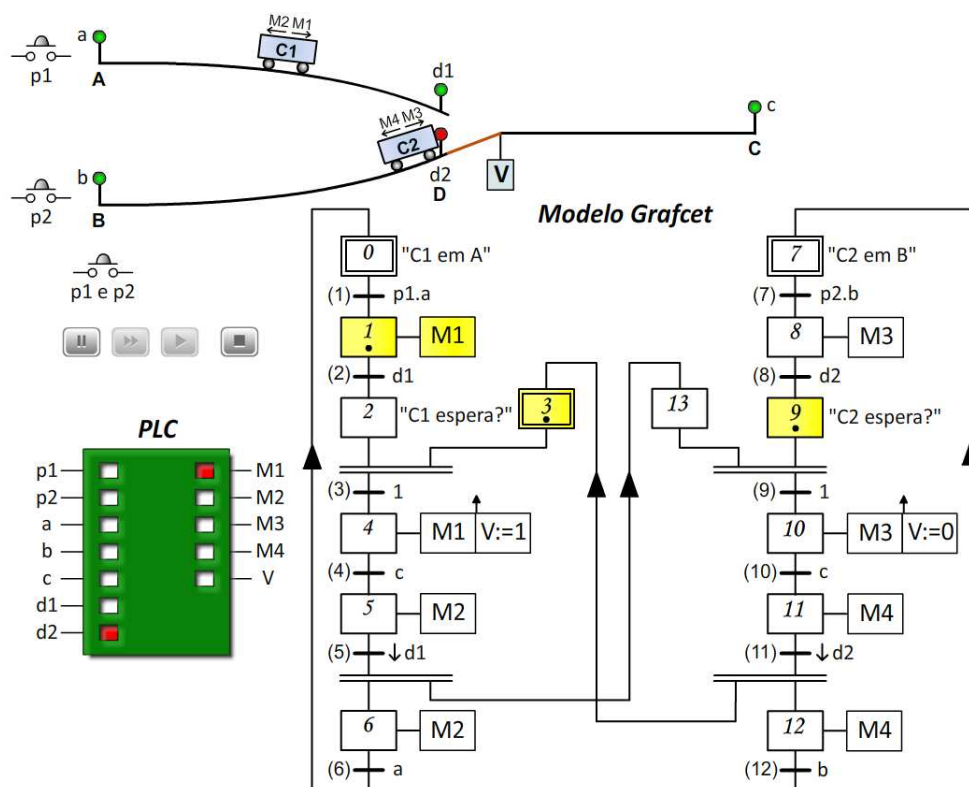


Figura 3.44 – Recurso partilhado - Exemplo 4.3.

activar o semáforo (etapa 13) que dá acesso ao carro 2. Nesta situação, o carro 1 só poderá voltar a utilizar o recurso partilhado após o carro 2 o utilizar, caso contrário vai esperar em *d1* (na entrada da zona de agulhagem).

3.4.5 Exemplo 5: Prioridades

O exemplo 5 [58] é referente à utilização de *prioridades*.

Enunciado do problema:

Um carro transporta os produtos de três linhas de produção independentes para um tapete de evacuação (ver Figura 3.45). O sistema é constituído por vários sensores: *ai* - detecta a presença de uma peça na linha *i*, com $i = 1, 2, 3$; *an* - detecta a presença de uma peça no carro de transporte; *aei* - detecta a presença de uma peça no início do tapete *evacuação*; *aef* - detecta a presença de uma peça no fim

do tapete evacuação; pi - fim de curso, indica o posicionamento do carro no final da linha i , com $i = 1, 2, 3$; pe - indica o posicionamento do carro no início da linha de evacuação. Os actuadores são os seguintes: E - desloca o carro para a esquerda; D - movimenta o carro para a direita; Ri - motor do tapete rolante i , com $i = 1, 2, 3$ (linhas), n (carro), e (evacuação); A capacidade de transporte do carro é de uma unidade e este encontra-se junto ao tapete de evacuação. Utilizando o *GRAFCET* vão ser modelados os controladores lógicos que correspondem às seguintes situações.

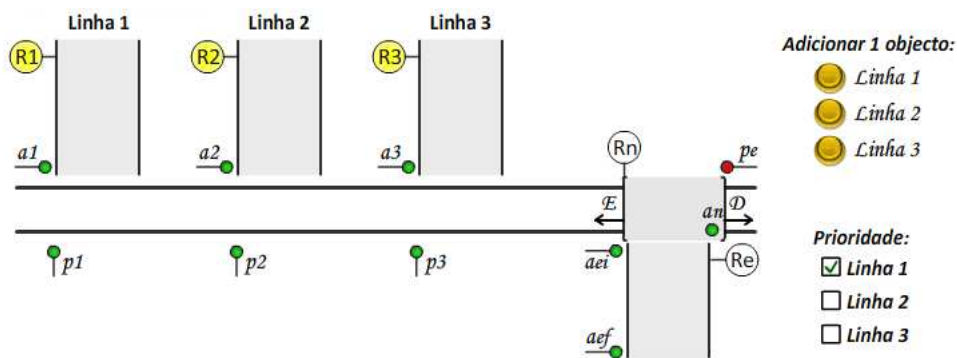


Figura 3.45 – Exemplo 5.

Exemplo 5.1

Enunciado: Inicialmente o carro e todos os tapetes estão vazios. A *linha 1* está em funcionamento. O motor pára quando é detectada uma peça pelo respectivo sensor de fim da linha $a1$ e vai aguardar até que o carro venha efectuar o seu transporte para a linha de *evacuação*. O mesmo se aplica às *linhas 2* e *3*. Ao premir o botão m o carro vai transportar uma peça. Caso não exista nenhuma peça a ser transportada o carro vai aguardar. A prioridade das linhas é atribuída da esquerda para a direita (inicialmente, linha 1 com maior prioridade e linha 3 com menor prioridade).

Este exemplo ilustra a aplicação de prioridades. O problema proposto é resolvido com a utilização de Grafcets parciais. Esta forma permite uma melhor compreensão e visualização no funcionamento das partes activas da animação (ver Figura 3.46).

No Grafcet do carro de transporte temos uma escolha de sequência que é definida

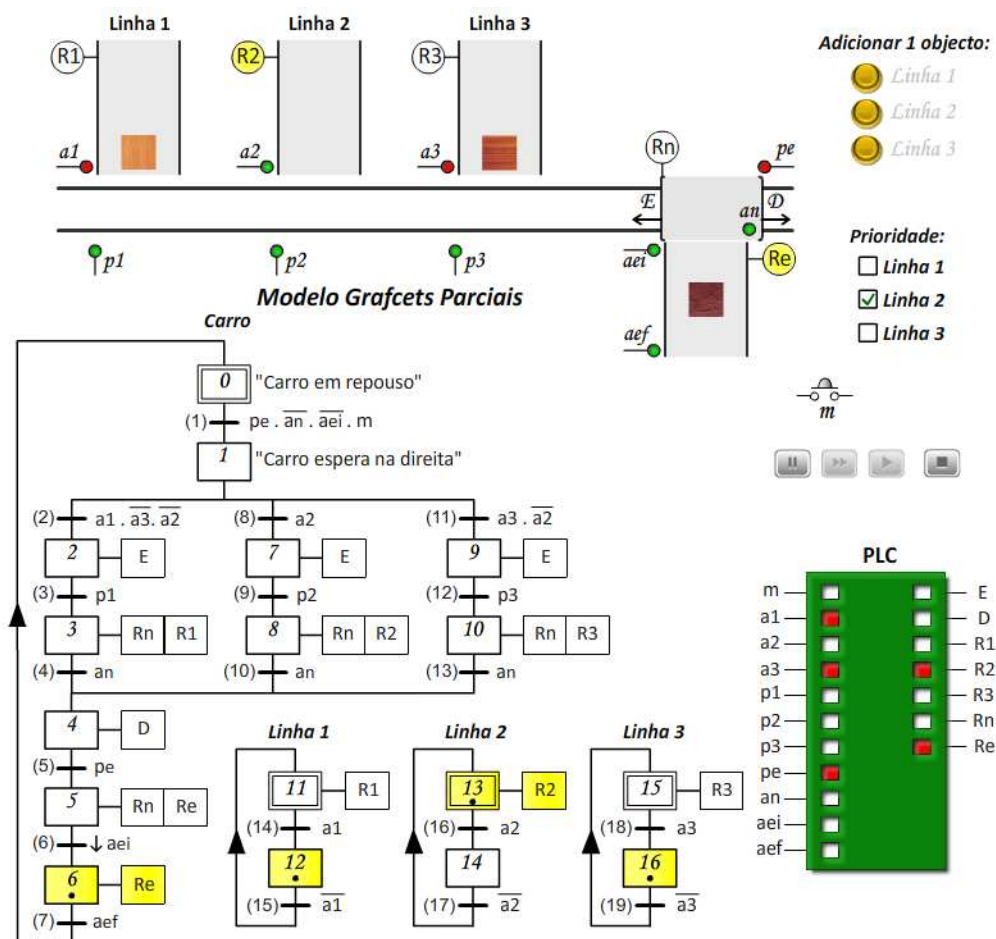


Figura 3.46 – Prioridades - Exemplo 5.1.

pelo uso de prioridades. Desta forma é estabelecida uma ordem para transportar os objectos das respectivas linhas, caso exista mais do que uma linha ocupada. Notar que as prioridades das sequências são definidas da esquerda para a direita. Isso mesmo pode ser observado pela alteração das receptividades associadas no início de cada uma das sequências.

Existem dois menus de opções disponíveis para o utilizador. O primeiro permite acrescentar um objecto numa das linhas escolhidas. O segundo menu permite estabelecer qual a linha com maior prioridade.

uma melhor compreensão e visualização do funcionamento das partes activas.

É elaborado um novo Grafcet parcial responsável por definir qual a linha com prioridade nesse momento. Com base neste Grafcet parcial e nos sensores de presença no final de cada linha de transporte são definidas funções Booleanas associadas às receptividades no início de cada sequência (transições 2, 8 e 11). A ordem de prioridade das linhas é da esquerda para a direita, ou seja, *linha 1* - *linha 2* - *linha 3* - *linha 1*... e assim sucessivamente. Para escrever as equações de cada uma das sequências consideram-se três situações distintas (prioridade de cada linha). Vamos tomar como exemplo a *linha 1*, para descrever essas situações:

1ª Situação: caso exista um objecto, a primeira condição é verificar se a linha tem prioridade sobre as outras pela etapa 17 activa ($X17=1$). Em caso afirmativo é executada a sequência da *linha 1* e a prioridade é atribuída à linha seguinte.

2ª Situação: existe um objecto na *linha 1* e a prioridade está definida para a linha 2 ($X18=1$). Nestas condições, temos de verificar se a *linha 2* e a *linha 3* não estão ocupadas, devido à ordem de prioridade das linhas mencionada anteriormente. Se alguma destas linhas (2 ou 3) estiver ocupada é executada a respectiva sequência. A Figura 3.47 ilustra este mesmo caso. No caso dessas linhas estarem livres, a sequência referente à *linha 1* é executada e a prioridade mantém-se na *linha 2*.

3ª Situação: existe um objecto na *linha 1* e a prioridade está definida para a *linha 3* ($X19=1$). Então, nestas condições, temos de verificar se a *linha 3* não está ocupada. Em caso afirmativo (*linha 3* livre), a linha seguinte com maior prioridade é a *linha 1*, razão pela qual não é necessário verificar se a *linha 2* está livre ou ocupada. Então a sequência referente à *linha 1* é executada e a prioridade mantém-se na linha 2. Pelo conjunto destas três situações obtemos como resultado a expressão: $u1 = a1.(X17 + X18.\overline{a2}.\overline{a3} + X19.\overline{a3})$. É aplicado o mesmo raciocínio para as funções $u2$ e $u3$.

3.4.6 Notas adicionais aos exemplos

Os exemplos propostos são didácticos tendo sido elaborados de forma a permitirem uma boa compreensão dos conceitos do GRAFCET. Existem situações nas simulações que ocorrem muito rapidamente e o utilizador pode não ter tempo suficiente para se aperceber dessas mudanças (disparos de transições, activação/desactivação de etapas, sinais do PLC, estado de sensores, etc.).

Como o e-GRAFCET é uma ferramenta didáctica e pretende-se transmitir com clareza todas as estruturas utilizadas nos exemplos bem como todos os elementos activos na animação, foram implementados botões que permitem ao utilizador controlar totalmente o decorrer da animação.

Em todas as animações existem botões de controlo auxiliares que possibilitam o seu controlo. Observando os botões da Figura 3.48, da esquerda para a direita, o primeiro botão permite parar a animação, o segundo botão permite avançar passo a passo, o terceiro botão permite retomar a execução normal e o quarto botão permite, em qualquer momento, inicializar toda a animação. Quando a simulação está parada, o utilizador é alertado por uma legenda colocada sob os botões. Este aviso permanece visível até que seja retomada a normal execução da animação.



Figura 3.48 – Botões de controlo da animação.

3.5 Testes sobre grafcet

De forma a avaliar os conhecimentos adquiridos é proposta a realização de vários testes, relativos ao GRAFCET, com respostas de escolha múltipla e de verdadeiro ou falso.

Os testes foram elaborados com o uso do software *Flash* e a linguagem de programação *ActionScript 2.0*. As perguntas estão guardadas e organizadas num ficheiro XML. Na secção seguinte é fornecida uma descrição dos pontos essenciais sobre a estrutura e funcionamento dos testes em linha.

3.5.1 Estrutura do ficheiro XML

No ficheiro XML cada questão é uma *tag*, e no interior desta são definidas cinco *tags*, uma para a pergunta propriamente dita e as restantes quatro contém opções de resposta a essa pergunta. Na *tag* da questão é declarado o atributo '*respCerta*' para definir qual a resposta certa à pergunta. De seguida é apresentada a estrutura de uma questão no XML:

```
<questao respCerta="2">
    <pergunta>Aqui é definida a pergunta.</pergunta>
    <resposta1>Opção de resposta 1;</resposta1>
    <resposta2>Opção de resposta 2;</resposta2>
    <resposta3>Opção de resposta 3;</resposta3>
    <resposta4>Opção de resposta 4.</resposta4>
</questao>
```

Para acrescentar uma pergunta ao ficheiro basta fazer uma cópia da estrutura anterior e alterar o conteúdo das *tags*.

3.5.2 Funcionamento e interface

Inicialmente, no *Flash*, é criada uma instância (classe) XML que carrega todo o conteúdo do ficheiro '*perguntas.xml*' ao mesmo tempo que é contabilizado o número total de questões. Com base neste número é preenchido um vector auxiliar com valores aleatórios diferentes (de 1 até o número total de questões). O objectivo é criar testes sempre diferentes cada vez que o aluno pretende testar os conhecimentos.

De seguida são preenchidos os vectores da pergunta e respostas com base nos valores do vector auxiliar. É definida uma variável '*actual*' que representa cada elemento

(nó) da classe XML previamente criada. Com base nos valores do vector auxiliar (aleatórios), é copiado o conteúdo da classe XML na posição indicada por esse valor, para os vectores associados às perguntas e respostas.

Após o correcto preenchimento destes vectores, toda a informação está devidamente organizada e o primeiro teste é iniciado. Cada teste é composto por 10 questões.

É apresentada uma questão de cada vez para responder, como ilustrado na Figura 3.49. A pergunta e respostas são carregadas dos respectivos vectores para caixas de texto dinâmicas. Para responder à questão, o aluno deve pressionar com o rato a resposta que pensa ser a correcta. Essa mesma resposta é indicada numa caixa de texto e armazenada num vector de respostas para, no final do teste, efectuar a análise das respostas correctas.

Avaliação de conhecimentos

5/10

Quantas regras definem o funcionamento do Grafcet?

3 regras;

4 regras;

2 regras;

► 5 regras.

Anterior

Seguinte

Resposta 4

Figura 3.49 – Testes online: exemplo de uma pergunta.

Para continuar a responder às perguntas basta premir o botão 'Seguinte'. O aluno pode sempre rever todas as perguntas e alterar as respostas. No final do teste, é feita uma validação afim de alertar o aluno para eventuais perguntas que estão por responder. De seguida, as respostas dadas são comparadas com as respostas correctas (indicadas pelo vector 'ok') e é apresentada toda a informação sobre a classificação. É indicada a percentagem obtida no teste e o respectivo nível atribuído, o número de respostas certas e erradas e ainda um pequeno rosto com estado contente ou triste, de acordo com a classificação (Figura 3.50).

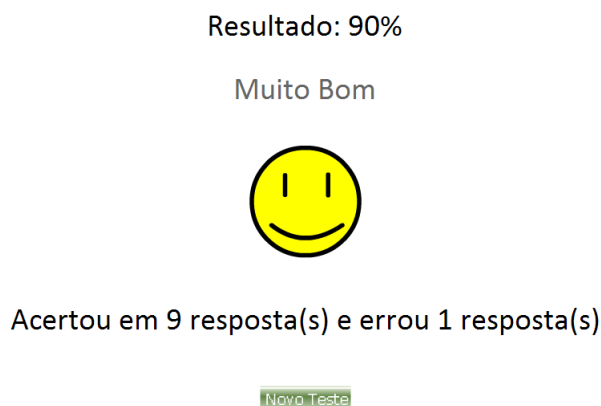


Figura 3.50 – Testes online: classificação.

O aluno pode efectuar a quantidade de testes que bem entender. Quando os testes forem todos respondidos a ordem das perguntas é novamente reformulada e são gerados novos testes.

3.6 Notas Finais

O presente capítulo foi dedicado à exposição do desenvolvimento do e-GR AFCET. Numa primeira fase foi realizada a descrição da estrutura da ferramenta bem como a justificativa das opções tomadas para a sua construção. De seguida, foi descrito e analisado todo o conteúdo que constitui a ferramenta. No seguimento deste capítulo, surge a necessidade de testar a aplicação desenvolvida, e-GR AFCET, no que concerne à sua utilização num ambiente real. O teste/validação desta ferramenta será apresentada no capítulo seguinte.

4

Validação do e-GR AFCET

4.1 Introdução

Este capítulo tem como objectivo descrever o teste e avaliação da ferramenta desenvolvida: e-GR AFCET. Para esse efeito, foi proposto a um conjunto de alunos, a realização de um inquérito. O objectivo do "Inquérito do e-GR AFCET" é sobretudo conhecer a opinião dos alunos em vários aspectos da ferramenta, de forma a avaliar na prática, a utilidade e o contributo para o ensino/aprendizagem, tendo em vista a sua melhoria e actualização.

4.2 Descrição do Inquérito

O inquérito apresentado no Anexo I é composto por um cabeçalho onde é feita a identificação do aluno, pelos campos: nome, número do aluno, curso e e-mail. A recolha de informação está dividida em três partes:

- A primeira parte é constituída por perguntas onde a resposta dada utiliza uma escala de apreciação da ferramenta, compreendida entre 1 (muito fraco) e 5 (muito bom).
- A segunda parte é constituída por questões com resposta afirmativa ou negativa, e também por algumas perguntas de desenvolvimento.

- A terceira parte contém um pequeno espaço reservado à descrição de eventuais erros encontrados na página ou nos conteúdos do e-GRAF CET. O aluno poderá incluir nesta secção alguns comentários pessoais e dar sugestões que contribuam para o melhoramento da ferramenta.

4.2.1 Implementação do Inquérito

Durante o desenvolvimento da ferramenta, esta foi utilizada como apoio às aulas da Unidade Curricular de *Automação e Controlo da Licenciatura em Engenharia das Energias* da UTAD no 2º semestre do ano lectivo 2008/2009. Além das aulas, os alunos tiveram a possibilidade de explorar e utilizar todos os recursos disponíveis na ferramenta fora da sala de aula, em casa por exemplo, dado o e-GRAF CET estar alojado num servidor, acessível em qualquer lugar com ligação à Internet¹.

No final do semestre o inquérito foi fornecido aos alunos da unidade curricular referida, bem como a alguns alunos da Unidade Curricular de *Automação Industrial do Mestrado em Engenharia Electrotécnica e de Computadores* que decorreu no 1º semestre do ano lectivo de 2008/2009. Esses inquéritos foram pré-analisados a fim de verificar o seu correcto preenchimento e validade como amostra para posterior análise.

4.3 Análise dos dados

4.3.1 Primeira Parte

As respostas dadas à primeira parte do questionário têm em conta uma escala de apreciação global da ferramenta, de 1 (muito fraco) a 5 (muito bom). É apresentada a questão e a representação gráfica da percentagem de respostas a essa mesma questão.

Questão 1: ” *Opinião geral sobre o e-GRAF CET*”

Como se pode observar do gráfico de barras apresentado na Figura 4.1, a avaliação

¹O e-GRAF CET está disponível em <http://al20905.com.sapo.pt/>

global da ferramenta foi positiva obtendo 64% de respostas no Bom e 29% no Muito Bom.

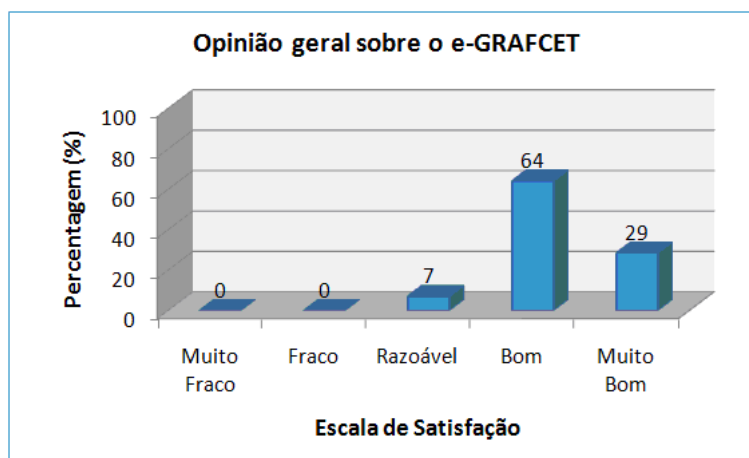


Figura 4.1 – Representação gráfica da percentagem de respostas à questão 1.

Questão 2: "Opinião sobre os conteúdos disponíveis"

Quando questionados sobre os conteúdos disponíveis pela ferramenta é atribuída, na grande maioria dos alunos, a classificação de Bom (86%). Uma menor percentagem considera-os Muito Bom (7%), o mesmo para a classificação de Razoável (7%) - Figura 4.2.

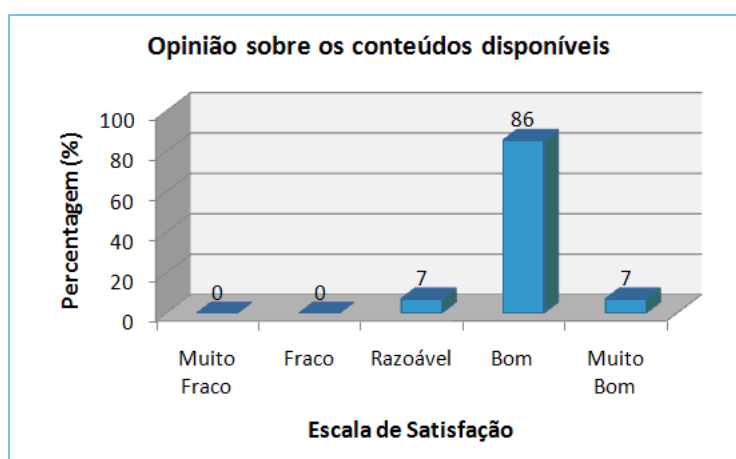


Figura 4.2 – Representação gráfica da percentagem de respostas à questão 2.

Questão 3: *” Como classifica a interface gráfica?”*

Os alunos são questionados quanto ao aspecto e interface gráfica da ferramenta. Como se pode observar pela Figura 4.3, as opiniões são equilibradas neste tópico, 43% atribuem à interface Muito Bom e uma percentagem ligeiramente menor classifica como Bom (36%). Os restantes alunos consideram a interface Razoável (21%).

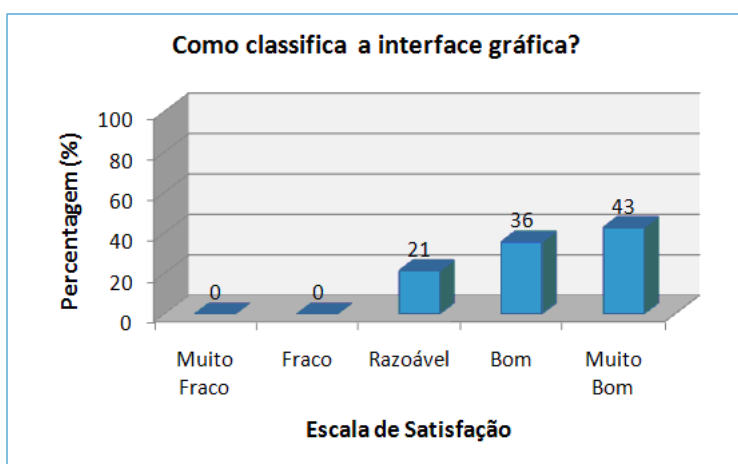


Figura 4.3 – Representação gráfica da percentagem de respostas à questão 3.

Questão 4: *” Como classifica a funcionalidade da página?”*

A funcionalidade da página é classificada, pela maioria dos alunos, como Bom (71%). Em menor percentagem responderam Muito Bom (7%) e os restantes classificam com Razoável (21%) - Figura 4.4.

Questão 5: *” Qual a classificação que atribui ao e-GRFCET como uma ferramenta útil no apoio ao ensino?”*

Os alunos quando questionados sobre a utilidade desta ferramenta no apoio ao ensino consideram-na como sendo boa (50%) e muito boa (29%). Os restantes consideram a sua utilidade Razoável (21%) - Figura 4.5.

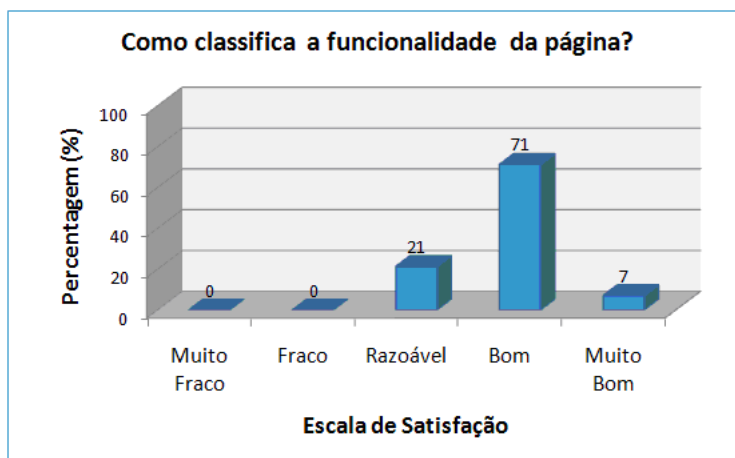


Figura 4.4 – Representação gráfica da percentagem de respostas à questão 4.

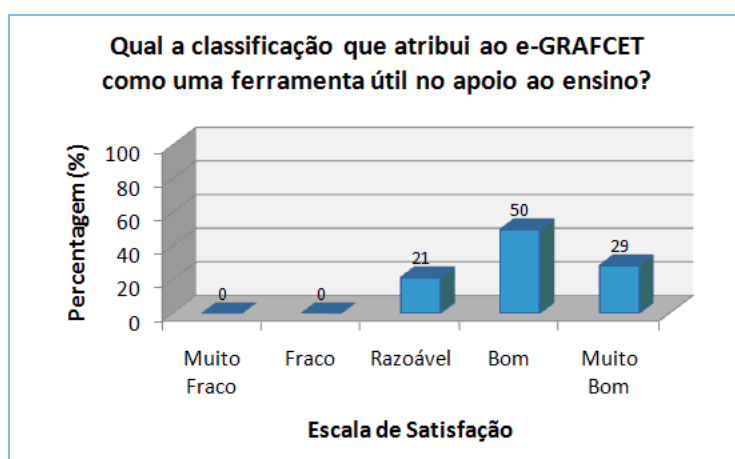


Figura 4.5 – Representação gráfica da percentagem de respostas à questão 5.

Questão 6: "Como classifica os exemplos práticos?"

Os exemplos práticos são uma das partes mais importantes desta ferramenta, pela simulação e demonstração prática dos conceitos teóricos. Os exemplos são classificados pela maioria dos alunos como Muito Bom (57%). A opinião dos restantes alunos considera Bom (29%) e Razoável (14%).

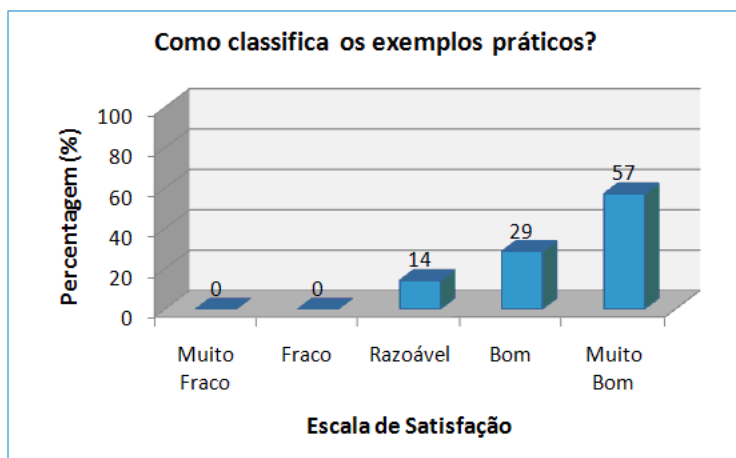


Figura 4.6 – Representação gráfica da percentagem de respostas à questão 6.

4.3.2 Segunda Parte

Esta parte é constituída por questões com resposta afirmativa (sim) ou negativa (não) e algumas perguntas de desenvolvimento.

Questão 7: "Os textos estão escritos de forma clara e concisa?"

Os textos foram escritos com grande cuidado e da forma mais clara possível para permitir ao aluno uma boa compreensão dos conceitos que se quer transmitir. Todos os alunos respondem afirmativamente quando questionados sobre a forma clara e concisa como os textos estão escritos.

Questão 8: "As animações Flash representam bem os vários conceitos do Grafcet?"

Como já foi mencionado no capítulo 3, as animações em Flash foram cuidadosamente planeadas e elaboradas de forma estratégica para representar o GRAFCET com clareza e sem ambiguidades. Todos os alunos afirmam que as animações em Flash representam bem os vários conceitos do GRAFCET.

Questão 9: "Esta ferramenta permite uma melhor compreensão do Grafcet?"

Do mesmo modo que as duas questões anteriores, todos os alunos afirmam que esta

ferramenta permite a melhor compreensão do GRAFCET.

Questão 10: " *Tem dificuldades em navegar no site?* "

Quando questionados sobre a navegabilidade no *site*, a grande maioria dos alunos inquiridos afirmou não ter qualquer tipo de dificuldade em aceder aos vários conteúdos disponíveis no *site* (Figura 4.7).

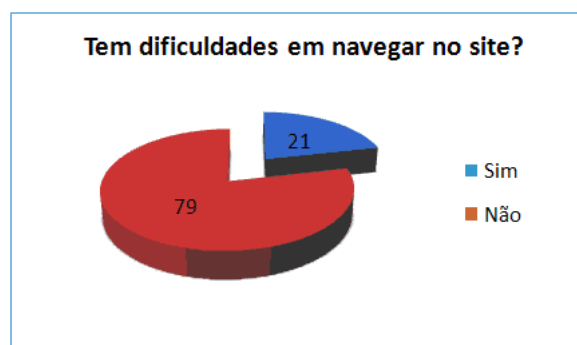


Figura 4.7 – Representação gráfica da percentagem de respostas à questão 10.

Questão 11: " *Em caso afirmativo, onde?* "

Esta questão é a continuação da anterior, para os casos afirmativos. De todos os alunos, três afirmaram ter alguma dificuldade relacionada com o acesso aos conteúdos. Por motivos de facilidade e uma constante actualização do e-GRAFCET, foi escolhido, inicialmente, um servidor temporário externo à UTAD. Esse servidor é bom para uso pessoal, mas por vezes provoca um pequeno atraso ao carregar a informação pela possível quantidade de tráfego a que está sujeito e do seu tempo de resposta, situação que tende a ocorrer à noite. Com um bom servidor não há motivo para isso acontecer tendo em conta a distribuição estratégica dos conteúdos e o reduzido espaço ocupado por estes no *site*. Esta situação será totalmente ultrapassada assim que o *site* esteja alojado nos servidores da UTAD.

Questão 12: *"Considera o software amigável?"*

Todas as respostas são afirmativas ao considerar o software desenvolvido amigável. Este é um ponto importante para o sucesso da ferramenta, permitir um uso fácil e agradável aos alunos.

Questão 13: *"Qual o conteúdo que mais gostou de ver representado?"*

Os alunos exprimiram que gostaram da ferramenta em geral, destacando os exemplos práticos, em particular a interação com o exercício permitida entre o utilizador no controlo da evolução de alguns exemplos. Também o paralelismo entre a simulação dos sistemas e dos modelos foi muito apreciada. Algumas opiniões consideram os exercícios práticos bastante chamativos e elucidativos, ajudam a esclarecer e a perceber melhor as várias situações propostas, o que vai permitir uma melhor aprendizagem. As estruturas do GRFCET também foram indicadas como bem representadas.

Questão 14: *"Algum conteúdo que não viu e gostava de ver representado?"*

Relativamente aos conteúdos que os alunos gostariam de ver representados, alguns responderam a associação entre Grafcet e os diagramas de contactos, implementação e codificação dos programas Grafcet para o PLC, e representar também as redes de Petri.

Questão 15: *"Ficou mais motivado para aprender o Grafcet?"*

Para finalizar esta segunda parte do inquérito, é questionado aos alunos se, depois de utilizarem a ferramenta, ficaram mais motivados para aprender o GRFCET, onde a resposta é unânime e afirmativa.

4.3.3 Terceira Parte

Esta parte é iniciada com um pequeno espaço reservado à descrição de erros na página ou nos conteúdos do e-GRFCET, encontrados no decorrer do uso da

ferramenta pelos alunos. De seguida os alunos expressaram alguns comentários pessoais e algumas sugestões de maneira a contribuir para o melhoramento da ferramenta.

Alguns erros encontrados pelos alunos foram levados em consideração e devidamente corrigidos.

Os comentários pessoais dos alunos são bastante positivos, consideram o e-GR AFCET amigável, apelativo, interessante, motivador na aprendizagem, onde fornece os conteúdos necessários para aprender o GR AFCET, todas as simulações ajudam a uma melhor compreensão, bem como as explicações e funcionalidades. Em resumo é uma boa ferramenta no apoio ao ensino e motiva o estudo do GR AFCET.

No decorrer do inquérito, a página inicial da ferramenta estava sem qualquer conteúdo demonstrativo, razão pela qual foi sugerido a sua melhoria. Neste momento encontra-se com uma animação inicial, onde é apresentado o conteúdo que pode ser encontrado no *site*, pela execução de um Grafcet. Deste modo, o aluno tem logo um primeiro contacto com uma série de noções e elementos do GR AFCET, mesmo não se apercebendo disso. Outra sugestão refere a diminuição do tempo entre as partes apresentadas. Este tempo de execução das animações, que em alguns casos poderá ser mais demorado, foi estabelecido propositadamente para permitir um bom acompanhamento e assimilação da informação que se pretende transmitir. Uma animação poderá, sim, tornar-se 'lenta' ao fim de várias utilizações, quando o aluno já sabe *à priori* o que vai acontecer.

A possibilidade de avaliar os conhecimentos pela realização de testes online não foi contemplada neste inquérito, dado ainda estar em fase de construção. No entanto, como a entrega dos inquéritos foi faseada e entretanto este recurso ficou operacional, alguns alunos tiveram a oportunidade de experimentar e expressar a sua opinião no inquérito. Afirmam ser um dos conteúdos que mais gostaram de ver no e-GR AFCET e consideram um bom utilitário na avaliação de conhecimentos.

4.4 Notas Finais

Este capítulo foi dedicado à validação do e-GRFCET por intermédio de um inquérito realizado a alguns alunos, os quais tiveram a oportunidade de experimentar a ferramenta e expressar a sua opinião.

De início, foram apresentados os objectivos do inquérito e dada a explicação de todo o seu conteúdo. Foi descrita a forma como este foi realizado aos alunos e de seguida, a informação recolhida pelos inquéritos foi analisada e apresentada.

Pela análise destes resultados verifica-se que os alunos consideram o e-GRFCET uma ferramenta amigável e bem construída sendo um bom contributo tanto no apoio às aulas, como no estudo individual de cada aluno.



Conclusão e Trabalho futuro

5.1 Síntese do Trabalho

Nesta "dissertação" foi apresentada uma ferramenta de apoio à aprendizagem do GRAFCET que permite complementar o seu ensino chamada e-GRAFCET. O trabalho realizado pretendeu, numa fase inicial, apresentar o GRAFCET como ferramenta de modelação de sistemas de eventos discretos. Foi apresentada uma revisão de alguns editores que facilitam o desenho do GRAFCET e software de automação que permitem elaborar programas em SFC. De seguida foram apresentadas ferramentas e sites existentes na Internet similares ao e-GRAFCET e demonstrado o interesse e importância no desenvolvimento de ferramentas deste género.

Numa segunda fase foram apresentados os desenvolvimentos e descrição da estrutura do e-GRAFCET assim como a justificação das opções tomadas para a sua construção, analisando todo o seu conteúdo. Na parte final deste trabalho foram apresentados os resultados de um inquérito realizado a vários alunos com o objectivo de testar e validar a ferramenta desenvolvida. Este inquérito pretendeu também verificar a utilidade e o contributo no ensino e aprendizagem dos alunos.

Todos os conteúdos presentes no e-GRAFCET são ilustrados com uma animação. As simulações estão bem animadas, e algumas permitem ainda acompanhar a evolução

e os sinais intervenientes com a ajuda de um cronograma, o que possibilita uma melhor compreensão.

Existe a possibilidade de testar os conhecimentos adquiridos pela realização de testes em linha (*online*) e ainda uma secção com diversas hiperligações (*links*) onde pode ser encontrada mais informação sobre este tópico na Internet.

O e-GR AFCET vai permitir um ensino mais dinâmico e motivador, sendo uma ferramenta útil à melhoria da qualidade do processo ensino/aprendizagem.

Apresenta-se, de seguida, algumas linhas de orientação para a continuação deste trabalho e o desenvolvimento de outros trabalhos.

5.2 Recomendações de Trabalho Futuro

O trabalho realizado no âmbito desta dissertação poderá ser complementado com novos desenvolvimentos, nomeadamente:

- Adicionar mais simulações e informação que permita abranger mais conteúdos do GR AFCET;
- Aumentar o número de exemplos, pela combinação de outras estruturas e noções, aplicadas a novas situações que se pretendem simular;
- Tradução da ferramenta para outras línguas, Inglês e Francês, por exemplo, abrangendo assim uma maior população estudantil, ultrapassando fronteiras;
- Criar outra componente didáctica onde o aluno interage com o Grafcet, modificando-o, podendo visualizar o resultado na simulação.

A realização dos próximos desenvolvimentos propostos é mais elaborada. Pode-se até afirmar que, alguns desses itens poderiam ser temas para uma Dissertação de Mestrado.

- Associação dos modelos em Grafcet na codificação para PLC;

- Adaptação do Grafcet para os diagramas de contactos;
- Desenvolver uma ferramenta semelhante para os diagramas de contactos;
- Desenvolver uma ferramenta semelhante para as Redes de *Petri*;
- Desenvolver um editor, por exemplo em Flash, onde seja possível manipular os vários elementos do Grafcet permitindo criar as próprias simulações.

5.3 Comentários Finais

Com a realização deste trabalho foi possível a consolidação de conhecimentos mais profundos no que respeita a *sistemas de controlo sequencial* e a *tecnologias de Internet*.

Em *sistemas de controlo sequencial* foram aprofundados conhecimentos ao nível de representação, edição e programação de tais sistemas através da utilização do GRAFCET.

Em *tecnologias de Internet*, foi possível adquirir conhecimentos em Flash e em programação Web, nomeadamente com a utilização das linguagens ActionScript, XML e HTML.

A ferramenta desenvolvida mostra de forma clara todos os conteúdos que a constituem. Por vezes, poderá existir ambiguidades ou dificuldades de compreensão, pelos alunos, da matéria leccionada nas aulas. Com o e-GRAFCET, o aluno tem a possibilidade de observar a evolução desse conteúdo com a respectiva explicação e simulação, com a vantagem de poder repetir mais do que uma vez até a sua completa compreensão.

No período de tempo em que os inquéritos foram respondidos a ferramenta encontrava-se ainda incompleta. Com o estado actual do e-GRAFCET, se os mesmos voltassem a ser respondidos, pelos alunos, a avaliação poderia ser ainda mais positiva.

Pelo bom resultado e aceitação dos alunos, tendo em conta as suas opiniões, o e-GRFCET é uma ferramenta de sucesso e de motivação para futura continuação e expansão dos seus conteúdos.

Em conclusão, pode afirmar-se que a realização deste trabalho foi uma tarefa árdua. Contudo é agradável e gratificante verificar que, perante o trabalho realizado e os resultados obtidos, se pode afirmar que foram cumpridos todos os objectivos propostos nesta dissertação.

Referências bibliográficas

- [1] R. David, “Grafcet: A powerful tool for specification of logic controllers,” *IEEE Transactions on Control Systems Technology*, vol. 3, pp. 253–268, September 1995.
- [2] J. Noivo and P. Garrido, “Elementos de automação,” Universidade do Minho. Departamento de Electrónica Industrial.
- [3] IEC (International Electrotechnical Commission), *IEC Standart 61131-3: Programmable Controllers - Part 3*, 1993.
- [4] IEC (International Electrotechnical Commission), *Language de spécification GRAFCET pour diagrammes fonctionnels en séquence*, 2 ed., 2000. Proposto para revisão de CEI 60848.
- [5] D. Keegan, *Foundations of Distance Education*. 3, Routledge, 1986.
- [6] C. P. Leão, P. M. Silva, P. C. Oliveira, N. C. Carvalho, and F. O. Soares, “Learning automation by doing it - a case-study,” *ICEE - International Conference on Enginneering Education*, 2007.

- [7] H. Guéguen and N. Bouteille, “Extensions of grafcet to structure behavioural specifications,” in *Control Engineering Practice 9*, pp. 743–756, PERGAMON, 2001.
- [8] S. Moreno and E. Peulot, *Le Grafcet, Conception – Implementation dans les Automates Programmables Industriels*. Paris: Educa Livre, 1998.
- [9] R. David and H. Alla, *Du Grafcet aux réseaux de Petri*. Hermes, 2 ed., 1992.
- [10] R. David and H. Alla, *Petri Net and Grafcet: Tools for modelling discrete event systems*. Prentice Hall, 1992.
- [11] P. Abati, “Le grafcet – description générale.” Website, 2003. <http://www.stielec.ac-aix-marseille.fr/cours/abati/grafcet1.htm>.
- [12] B. S. Torres, D. G. Santos, and M. O. Fonseca, “Implementação de estratégias de controlo multimalha utilizando a norma 61131-3 e ferramentas de pims,” 2003.
- [13] M. Öhman, S. Johansson, and K.-E. Årzén, “Implementation aspects of the PLC standard IEC 1131-3,” in *Control Engineering Practice 6*, pp. 545–555, PERGAMON, 1998.
- [14] J. R. C. Pinto, *Técnicas de Automação*. Lidel, 2004.
- [15] J. Novais, *Programação de Autómatos – Método GRAFCET*. Lisboa: Fundação Calouste Gulbenkian, 2004.
- [16] Adepa/Afcet, *Le Grafcet*. França: Cépaduès Éditions, 2 ed., 1992.
- [17] A. M. S. Francisco, *Autómatos programáveis*. Lidel, 2003.
- [18] P. M. Oliveira and E. S. Pires, *Introdução ao Grafcet*. No. 148, Universidade de Trás-os-Montes e Alto Douro, Vila Real: Série Didáctica no Ciências Aplicadas.

- [19] C. A. Couto and J. Fonseca, “Editor gráfico de grafcet,” in *3^as Jornadas Hispano Lusas de Ingeniería Eléctrica*, 1993.
- [20] S. Di-Meglio, “Sfcedit.” Website.
<http://stephane.dimeglio.free.fr/sfcedit/index.html>.
- [21] “Graftor.” <http://www.naskita.com/linux/graftor/graftor.shtml>.
- [22] “Editsab.” Lycée Antoine Bourdelle, Montauban.
<http://vrous.club.fr/index.html>.
- [23] “Machcet.” <http://home.tele2.fr/fr-25596/page4.html>.
- [24] “Wingrafcet.” http://pagesperso-orange.fr/jmdefais/techn_jm/wingraf.htm.
- [25] M. Ribas, “Grafcet,” 2008. <http://admr.cad.free.fr/>.
- [26] “Automation studio.” <http://www.automationstudio.com/EDUC/index.htm>.
- [27] J. Machado and E. Seabra, “Competency-based education in automation teaching,” 2003.
- [28] “Codesys.” The IEC 61131-3 programming system.
http://www.3s-software.com/index.shtml?en_download.
- [29] “Isagraf.” <http://www.isagraf.com/>.
- [30] COPALP, “Straton workbench.” <http://www.copalp.com/>.
- [31] J. L. Díez, A. Valera, J. L. Navarro, M. Vallés, and A. Encinas, “An interactive course on logic controllers design using grafcet,” 2006.
- [32] F. B. Santos, “Curso multimedia de grafcet adaptado para discapacitados visuales.” Website. http://personales.upv.es/jldiez/curso_grafcet/.
- [33] F. O. Soares, C. P. Leão, A. Moreira, A. Mota, F. Arantes, and A. Dias, “Funcionalidades web como complemento ao ensino presencial,” 2004.

- [34] C. Leão and F. O. Soares, “Internet based complementary learning and assessment,” *Current Developments in Technology-Assisted Education*, pp. 504–508, 2006.
- [35] F. Arantes and F. Dias, “Automação – DEI – universidade do minho.” Website. <http://dei-s1.dei.uminho.pt/lic/AUT/index.html>.
- [36] C. Leão and F. O. Soares, “Application of new methodologies in an industrial electronics engineering course: case study,” Chapter 28.
- [37] F. Estragués. Website. <http://perso.wanadoo.es/kiko2000/entrada.es.html>.
- [38] P. Trau. http://www-ipst.u-strasbg.fr/pat/autom/grafce_t.htm.
- [39] P. Berger. Website. http://philippe.berger2.free.fr/le_grafcet.htm.
- [40] P. Abati, “Le grafcet - description générale.” Website. <http://www.stielec.ac-aix-marseille.fr/cours/abati/grafcet1.htm>.
- [41] “Tp line, les travaux pratiques de sciences industrielles sur des systèmes en fonctionnement en temps réel.” Website. <http://www.tpline.net/cours/3automatique/grafcet/html/grafcet001.php>.
- [42] R. Papanicola. Website. <http://robert.papanicola.free.fr/spip/spip.php?rubrique14&&style=1>.
- [43] “Sciences et techniques industrielles.” Website. http://www.tn.refer.org/hebergement/cours/sys_disc/Grafcet_acceuil.html.
- [44] TecAtlant, “Introduction au grafcet.” Website. <http://www.tecatlant.fr/grafcet/>.
- [45] O. Boix, “Curso de grafcet y gemma.” Website, 2002. <http://edison.upc.edu/curs/grafcet/>.

- [46] “Le grafcet, système: Perceuse automatique.” Website.
<http://www.edunet.tn/ressources/sitetabl/sites/sousse/lbba/Grafcet.html>.
- [47] I. N. des Sciences Appliquées et de Technologie, “Cours sur les systèmes à événements discrets - le grafcet.” Website.
http://www.tn.refer.org/hebergement/cours/sys_disc/Grafcet_acceuil.html.
- [48] L. B. Gouveia, “O e-learning para suporte ao ensino presencial universitário,” *Repensar os Media: Novos Contextos da Comunicação e da Informação*, 20 e 21 Outubro 2005. Livro de actas - 4º SOPCOM, Universidade de Aveiro.
- [49] D. Leal and L. Amaral, “Do ensino presencial ao e-learning,” Universidade do Minho, Braga, Portugal.
- [50] J. da Silva Maia, “Ferramenta para aprendizagem de automação e controle discreto: Controlador lógico programável (plc) na modalidade e-learning,” Master’s thesis, Florianópolis, 2002.
- [51] M. J. Gomes, “Desafios do e-learning: Do conceito às práticas,” in *Actas do VIII Congresso GalaicoPortuguês de PsicoPedagogia*, (Braga: CIEd / IEP / UM), pp. 66–76, CD-Rom, 2005.
- [52] G. E. McCray, *The hybrid course: Merging on-line instruction and the traditional classroom*, vol. 1. Springer Netherlands, March 2000.
- [53] M. Balazinski and A. Przybylo, “Teaching manufacturing processes using computer animation,” *Journal of Manufacturing Systems*, vol. 24, no. 3, pp. 237–243, 2005.
- [54] I. Macromedia, *Aprendendo ActionScript 2.0 no Flash*. 2005.
- [55] S. Bhangal and K. Besley, *Foundation Flash 8*. Friends of ED, 2005.
- [56] W. Schools, “Xml introduction - what is xml?.” Website.
http://www.w3schools.com/xml/xml_what_is.asp.

- [57] K. Besley, S. Bhangal, D. Powers, and E. Dolocki, *Foundation ActionScript for Flash 8*. Friends of ED, 2006.
- [58] D. Chappe, “Cours grafcet.” Website.
http://www.lab.ens2m.fr/cours_automatique/grafcet/grafcet.htm.



Inquérito do e-GRAFCET

INQUÉRITO DO e-GRFCET

Estes dados destinam-se a uma avaliação da ferramenta e-GRFCET, tendo em vista a sua melhoria/actualização.

Identificação

Nome: N.º de Aluno:

Curso: E-mail:

Recolha de Informação

1. Classifique as seguintes afirmações, tendo em conta a seguinte escala de satisfação:

1 (Muito Fraco) 2 (Fraco) 3 (Razoável) 4 (Bom) 5 (Muito Bom)

- Opinião geral sobre o *e-GRFCET*:(1-5) _____
- Opinião sobre os conteúdos disponíveis:(1-5) _____
- Como classifica a interface gráfica?(1-5) _____
- Como classifica a funcionalidade da página?(1-5) _____
- Qual a classificação que atribui ao *e-GRFCET* como uma ferramenta útil no apoio ao ensino?(1-5) _____
- Como classifica os exemplos práticos?.....(1-5) _____

2. Responda às seguintes questões:

s (sim) / n (não)

- Os textos estão escritos de forma clara e concisa?(s/n) _____
- As animações Flash representam bem os vários conceitos do grafcet?
.....(s/n) _____
- Esta ferramenta permite uma melhor compreensão do Grafcet?
.....(s/n) _____
- Tem dificuldades em navegar no site? (s/n) _____
- Em caso afirmativo, onde? _____

- Considera o software amigável? (s/n) _____
- Qual o conteúdo que mais gostou de ver representado?

- Algum conteúdo que não viu e gostava de ver representado?

- Ficou mais motivado para aprender o Grafcet?(s/n)_____

Utilize o seguinte quadro para descrever algum erro que possa ter encontrado.

| Página: | Descrição: |
|---------|------------|
| | |
| | |

Comentários Pessoais:

Sugestões:

Nota: O e-GRFCET ainda está em construção!

Obrigado pela participação!