



**INSTITUTO
FEDERAL**
Catarinense

Instituto Federal Catarinense
Bacharelado em Engenharia de Controle e Automação
Campus Luzerna

João Peterson Scheffer

**Desenvolvimento de um método e sistema para trans compilação e simulação
de redes de petri para utilização em controladores lógicos industriais**

Luzerna
Dezembro de 2022

João Peterson Scheffer

**Desenvolvimento de um método e sistema para trans compilação e simulação
de redes de petri para utilização em controladores lógicos industriais**

Trabalho de conclusão apresentado à banca examinadora do curso de Bacharelado em Engenharia de Controle e Automação, do Instituto Federal Catarinense, Campus Luzerna, como requisito para a obtenção do título de Bacharel em Engenharia de Controle e Automação, em cumprimento às exigências de sua componente curricular. Orientador: Prof. Msc. Tiago Javorani Prati

Luzerna
Dezembro de 2022

João Peterson Scheffer

**Desenvolvimento de um método e sistema para trans compilação e simulação
de redes de petri para utilização em controladores lógicos industriais**

Trabalho de conclusão apresentado à banca examinadora do curso de Bacharelado em Engenharia de Controle e Automação, do Instituto Federal Catarinense, Campus Luzerna, como requisito para a obtenção do título de Bacharel em Engenharia de Controle e Automação, em cumprimento às exigências de sua componente curricular.

Luzerna (SC), 08 de dezembro de 2022:

Prof. Msc. Tiago Javorani Prati
Instituto Federal Catarinense

BANCA EXAMINADORA

Prof. Msc. XXXXXXX
Instituto Federal Catarinense

Prof. Dr. XXXXXXX
Instituto Federal Catarinense

Exemplo: Este trabalho é dedicado aos meus colegas de classe, aos meus queridos pais e professores

AGRADECIMENTOS

Inserir os agradecimentos aos colaboradores à execução do trabalho.

Texto da Epígrafe. Citação relativa ao tema do trabalho. É opcional. A epígrafe pode também aparecer na abertura de cada seção. Deve ser elaborada de acordo com a NBR 10520.

(SOBRENOME do autor da epígrafe, ano)

RESUMO

O texto do resumo em língua portuguesa deve ser digitado em um único bloco, sem espaço de parágrafo. O resumo deve ser significativo, composto de uma sequência de frases concisas, afirmativas e não de uma enumeração de tópicos. Não deve conter citações. O espaçamento entre linhas é 1,5 e o tamanho da fonte é 12. Abaixo do resumo deve-se informar as palavras-chave (palavras ou expressões significativas retiradas do texto) ou, termos retirados de thesaurus da área. De 150 a 500 palavras.

Palavras-chave: redes de petri; automação industrial; software. .

ABSTRACT

O texto do resumo em língua inglesa deve ser digitado em um único bloco, sem espaço de parágrafo. O resumo deve ser significativo, composto de uma sequência de frases concisas, afirmativas e não de uma enumeração de tópicos. Não deve conter citações diretas. Abaixo do resumo deve-se informar as palavras-chave (palavras ou expressões significativas retiradas do texto) ou, termos retirados de thesaurus da área. Até 500 palavras.

Keywords: petri nets; industrial automation; software. .

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

CÓDIGOS

LISTA DE ABREVIATURAS E SIGLAS

PLC	<i>Programmable logic controller</i>
CLP	Controlador lógico programável
WYSIWYG	<i>What you see is what you get</i>
Vscode	<i>Visual studio code</i>

SUMÁRIO

1	INTRODUÇÃO	25
1.1	JUSTIFICATIVA	26
1.2	OBJETIVOS	27
1.2.1	Objetivo geral	27
1.2.2	Objetivos específicos	27
2	METODOLOGIA	29
3	FUNDAMENTAÇÃO TEÓRICA	31
3.1	APLICAÇÃO GRÁFICA	31
4	DESENVOLVIMENTO	33
5	RESULTADOS	35
6	CONCLUSÃO	37
	REFERÊNCIAS	39
	APÊNDICE A – EXEMPLO DE APÊNDICE	41
	ANEXO A – EXEMPLO DE APÊNDICE	43

1 INTRODUÇÃO

Quando se trata de sistemas dinâmicos como um todo há sempre um interesse de criar, entender, prever e controlar estes sistemas, e no âmbito de sistemas discretos orientados a eventos ¹, como alguns sistemas industriais de automação de interesse neste trabalho, há várias metodologias, práticas, tecnologias e linguagens de programação capazes de trabalhar estes sistemas de forma a alcançar resultados esperados, cada qual possui vantagens e desvantagens.

No âmbito industrial, para realizar o controle e automação de processos, comumente é utilizado controladores lógicos programáveis, PLC's, e estes podem ser programados para realizar as funções desejadas e para isso empregam a utilização de linguagens de programação como:

- *Ladder*, que representa um processo de forma visual inspirada em lógica de contatos ².
- Lista de instrução, que é uma linguagem escrita, não visual, e que representa um fluxo de operações com base em comandos de texto.
- Grafcet, uma linguagem visual que representa um processo em forma de fluxo com base em passos de um nó para outro nó do processo, onde cada nó é uma instrução de ação e cada passo é dado conforme uma transição atrelada a um evento.

Ainda há outras tecnologias que possuem outras formas abstratas e abordagens diferentes, sendo a linguagem *Ladder* um exemplo de aproveitamento de conhecimento e simplicidade, porém há casos onde certas ferramentas não apresentam melhor desempenho e eficiência dado certos tipos de especificações, em especial no caso do *Ladder*, onde controle de estado ³ e paralelismo ⁴ são conceitos difíceis de serem implementados.

Para resolver algum destes problemas, condições e situações como as mencionadas, emprega-se o uso de linguagens de modelagem lógica capazes de definir um sistema desejado, representa-lo como um sistema discreto orientado a eventos, onde podem-se empregar abstrações lógicas adequadas a situação ou ainda prover ferramentas para realizar análise como alcançabilidade, significando se existem casos passíveis de serem realizados apesar da intenção contrária do projeto, se o sistema é reinicializável, significando a possibilidade do sistema de retornar a um estado inicial sempre, dentre outras análises complexas.

Neste contexto há várias linguagens de modelagens lógicas, como autômatos, máquinas de *Moore*, dentre outras linguagens lógicas, que podem ser usadas para abstrair alguns

¹ Sistemas que trabalham com valores discretos como ligado e desligado por exemplo e possuem mudança de estado conforme eventos também discretos.

² Uma forma simples de programação baseada na representação visual de lógica de contato, como vista em diagramas com relés e contadoras.

³ Armazenamento e lógica do estado/situação atual de um processo

⁴ Capacidade de unir dois fluxos de trabalho com razões de trabalho diferentes em um ponto definido.

conceitos como os citados anteriormente e nesse contexto apresenta-se então a rede de Petri (IZHIKEVICH, 2011), que é capaz de reproduzir conceitos abstratos como paralelismo, sincronia, mutualidade exclusiva, etc., de forma simples bem como vários outros conceitos comuns de lógica e aritmética e conceitos ainda exclusivos, intrínsecos a si própria.

As redes de Petri são assim um instrumento de maior formalidade e podem representar conceitos abstratos, fazendo da mesma uma ótima ferramenta para automação e controle de sistemas a eventos discretos, e que cada vez mais vem sendo estudada em meio acadêmico quanto a sua utilização como mecanismo de modelagem de processos, modelagem de sistemas críticos (LEVESON; STOLZY, 1987) e (GHEZZI et al., 1991), e prevenção de *deadlocks* ⁵ (KAID et al., 2015).

Visto o um grau de interesse acadêmico, aplicações em situações bem definidas, significando-se tratar de uma ferramenta adequada, e ainda a utilidade industrial, dado sua ótima representação de processos e liberdade de abstração, pode-se dizer que há interesse prático em redes de Petri, porém o emprego real deste tipo de tecnologia é mínimo, existem metodologias (MOREIRA; BASILIO, 2014) que por exemplo, propõem a funcionalidade de interpretação de redes de Petri de alto nível para código *Ladder* capaz de ser então usado em PLC's para uso industrial, função implementada em projetos como *PetriLab* (SOUZA, 2015), porém não trazem integração real, uma boa experiência de programação ou mesmo boas práticas de design de *software*, características que são valorizadas e por vezes indispensáveis para o processo de desenvolvimento, integração, teste, manutenção e melhoria contínua de sistemas e controle de sistemas em meio industrial, em outras palavras, há necessidade de ferramentaria adequada e atrativa às empresas e desenvolvedores destes sistemas industriais bem como difusão da utilização de redes de Petri como ferramenta de uso geral e específico.

1.1 JUSTIFICATIVA

Sendo apresentado o estado atual de disseminação de utilização de redes de Petri, tanto geral como industrial, justifica-se a criação de ferramentaria necessária ao desenvolvimento de aplicações. Aplicações tais que implementam necessariamente formas visuais de representação e edição, bem como simulação e observação do funcionamento destas redes e ainda a funcionalidade de transformar estas redes em programas prontos para utilização em meio industrial.

A capacidade de transformar tais redes em programas industriais, compreende-se como o processo de transcompilação, onde a rede de Petri é traduzida através de um algoritmo proposto para o resultado de saída, sendo esta saída código puro, arquivos digitais ou programas prontos para utilização com PLC's industriais.

⁵ Singularidade em processos onde o sistema entra em um ponto de parada de forma imprevista e não possui forma de autocorreção, permanece ou parado em falha ou em repetição contínua de uma única instrução.

1.2 OBJETIVOS

1.2.1 Objetivo geral

Desenvolvimento de um ambiente de edição, simulação e compilação de rede de petri voltadas para utilização em meio industrial, PLC's.

1.2.2 Objetivos específicos

- Desenvolvimento de uma biblioteca implementada em linguagem C, contendo uma estrutura de dados que represente a rede de petri em memória e que implemente a dinâmica de funcionamento, a simulação.
- Desenvolvimento de algoritmos de trans compilação de redes de petri para o seguinte alvo:
 - Lista de instrução, em formato de texto.
- Desenvolvimento de uma aplicação gráfica para edição e simulação de redes de petri utilizando o *framework* Flutter com linguagem de programação Dart.
- Desenvolvimento de uma biblioteca em Dart que porta as funções da biblioteca em linguagem C para utilização dentro do *framework* Flutter.

2 METODOLOGIA

Este trabalho será de grande parte um processo de desenvolvimento por parte do autor de maneira autônoma, dado que os tópicos de programação e alguns detalhes de implementação são por natureza de livre implementação, tornando este trabalho por grande parte um trabalho exploratório. Em contraste com os métodos utilizados, as visões e referências de trabalhos passados, que irão servir de guia para implementação de alguns aspectos gerais, são de cunho bibliográfico.

No que se refere ao aspecto visual da aplicação de edição, foi tomado inspiração do trabalho anterior Petrilab (SOUZA, 2015) bem como o editor de texto Visual Studio Code (MICROSOFT, 2022). De forma mais ampla, redes de petri sempre possuíram um visual distinto e difundido pela literatura, o que se traduz na implementação do Petrilab e eventualmente deste trabalho. Quanto a definição do tipo de rede de petri adotada neste trabalho bem como a base de desenvolvimento dos algoritmos de trans compilação, estes foram baseados em revisão bibliográfica e trabalhos anteriores.

3 FUNDAMENTAÇÃO TEÓRICA

3.1 APLICAÇÃO GRÁFICA

Editores visuais são provavelmente um dos tipos de aplicações gráfica mais comuns, pois conforme a evolução do computadores pessoais foi avançando, também foram as formas de interação, desde os consoles e telas puramente em texto para hoje dia interfaces e sistemas operacionais graficamente ricos. Em especial o tipo de programa WYSIWYG, do inglês, o que você é o que realmente é, ou seja, o que você interage na tela é realmente como é o produto final, contraste com aplicações como o sistema \LaTeX (LATEX, 2022), onde o resultado de saída, um arquivo PDF, não é o mesmo que foi editado, um arquivo de texto, sendo um exemplo de aplicação WYSIWYG o Libreoffice Writer (LIBREOFFICE, 2022), um editor de texto capaz de realizar formatação, assim como o \LaTeX , mas onde você digita e imediatamente vê o resultado.

Neste contexto as aplicações é

4 DESENVOLVIMENTO

5 RESULTADOS

Este *template* contém algumas seções criadas na tentativa de facilitar seu uso. No entanto, não há um limite máximo ou mínimo de seção a ser utilizado no trabalho. Cabe a cada autor definir a quantidade que melhor atenda à sua necessidade.

6 CONCLUSÃO

As conclusões devem responder às questões da pesquisa, em relação aos objetivos e às hipóteses/pressupostos. Devem ser breves, podendo apresentar recomendações e sugestões para trabalhos futuros.

REFERÊNCIAS

- GHEZZI, C. et al. A unified high-level petri net formalism for time-critical systems. *IEEE Transactions on software engineering*, IEEE Computer Society, v. 17, n. 2, p. 160, 1991.
- IZHIKEVICH, E. M. *Petri Net*. 2011. Disponível em: http://www.scholarpedia.org/article/Petri_net.
- KAID, H. et al. Applications of petri nets based models in manufacturing systems: A review. In: *Proceedings of the International Conference on Operations Excellence & Service Engineering, Orlando, FL*. [S.l.: s.n.], 2015. p. 516–28.
- LATEX. *The Latex project*. 2022. Disponível em: <https://www.latex-project.org>. Acesso em: 8 dez. 2022.
- LEVESON, N.; STOLZY, J. Safety analysis using petri nets. *IEEE Transactions on Software Engineering*, SE-13, n. 3, p. 386–397, 1987.
- LIBREOFFICE. *Writer*. 2022. Disponível em: <https://www.libreoffice.org/discover/writer/>. Acesso em: 8 dez. 2022.
- MICROSOFT. *Visual Studio Code*. 2022. Disponível em: <https://code.visualstudio.com>. Acesso em: 8 dez. 2022.
- MOREIRA, M. V.; BASILIO, J. C. Bridging the gap between design and implementation of discrete-event controllers. *IEEE Transactions on Automation Science and Engineering*, v. 11, n. 1, p. 48–65, 2014.
- SOUZA, A. L. de. *PETRILAB: UMA PLATAFORMA PARA SIMULAÇÃO E GERAÇÃO DE DIAGRAMAS LADDER DE CONTROLADORES A EVENTOS DISCRETOS MODELADOS POR REDES DE PETRI*. Tese (Doutorado) — Departamento de Engenharia Elétrica da Escola Politécnica, Universidade , 2015.

APÊNDICE A – EXEMPLO DE APÊNDICE

Textos elaborados pelo autor, a fim de completar a sua argumentação. Deve ser precedido da palavra APÊNDICE, identificada por letras maiúsculas consecutivas, travessão e pelo respectivo título. Utilizam-se letras maiúsculas dobradas quando esgotadas as letras do alfabeto.

ANEXO A – EXEMPLO DE APÊNDICE

São documentos não elaborados pelo autor que servem como fundamentação (mapas, leis, estatutos). Deve ser precedido da palavra ANEXO, identificada por letras maiúsculas consecutivas, travessão e pelo respectivo título. Utilizam-se letras maiúsculas dobradas quando esgotadas as letras do alfabeto.