ATUALIZAÇÃO E EXPANSÃO DE SOFTWARE PARA PROGRAMAÇÃO BASEADA NA IEC 61499

Mauricio Martins Taques Filho, Yuri Kaszubowski Lopes, André Bittencourt Leal, Herbert Albérico De Sá Leitão, Roberto Silvio Ubertino Rosso Junior

INTRODUÇÃO

Com o avanço da indústria 4.0 no setor da automação industrial, os processos automatizados vêm sendo cada vez mais autônomos e complexos, e por sua vez, aumentando a margem de falhas. A Comissão Eletrotécnica Internacional (IEC) desenvolveu uma norma específica, a IEC 61499 [1, 2, 3], que define como blocos de função (FB) podem ser utilizados em sistemas distribuídos de processo industrial, medição e controle [4]. Nesse paradigma, por meio de suas propriedades fundamentais, como a interoperabilidade, portabilidade e configurabilidade, a norma atua como uma solução a essa problemática, trazendo consigo uma arquitetura estável, reutilizável, e que, não apenas reduz a complexidade como também traz uma forma simplificada de modelar sistemas de automação distribuídos.

Na atualidade, existem poucas ferramentas para desenvolvimento aderente à norma IEC 61499, diante disso, em 2010, o GASR (Grupo de Automação de Sistemas e Robótica) desenvolveu o FBE (Function Block Environment), ambiente de execução, criação e edição de blocos de função escrito em Lua. Porém, devido a descontinuação do projeto, foi necessária uma atualização do software para ferramentas de desenvolvimento mais modernas.

Assim, em 2023, o projeto foi reformulado em Python, mantendo apenas algumas das funcionalidades da ferramenta original e vem evoluindo desde então. Este projeto é uma continuidade do trabalho de 2024/2025, e como prosseguimento do projeto houve a necessidade da expansão da aplicação abrangendo por exemplo importações de autômatos, simulação de redes de blocos de função e migração da biblioteca gráfica.

DESENVOLVIMENTO

A biblioteca para desenvolvimento de interface gráfica GTK (GIMP *ToolKit*) é frequentemente usada na implementação de aplicações desenvolvidas para o GNOME, ambiente de trabalho gráfico usado em sistemas operacionais baseados em Linux. O FBE 3 inicialmente aderiu ao GTK 4 [5], o qual oferecia ferramentas gráficas atualizadas para GNOME, contudo, a portabilidade de uma aplicação implementada sob GTK 4 para outros sistemas operacionais como Windows ou MacOS não era intuitivo e amigável ao usuário. Visto que a norma IEC 61499 é majoritariamente usada por profissionais da automação industrial. E que, muitas vezes, não são usuários de sistemas Linux, uma decisão de projeto foi abandonar o uso da GTK 4 em favor de outra solução gráfica.

Logo, a interface gráfica do projeto passou a ser implementada em PyQt 6 [6], uma biblioteca gráfica multiplataforma que respeita a propriedade de portabilidade da norma. Na refatoração do código fonte do FBE já existente, as classes que definem os tipos de blocos de função e o algoritmo de *parser* que transforma FB definidos em um arquivo XML (*eXtensible Markup Language*) para objetos dentro do código foram mantidos. O restante das classes de GUI (*Graphical User Interface*) foram traduzidas de GTK para PyQt.

O mapeamento de eventos para algoritmos é expresso por meio de uma notação especial de transição de estados denominada ECC (*Execution Control Chart*) [4], que é uma máquina de estados finitos (Autômato Finito). Na versão do FBE desenvolvida no período 2024/2025, foi implementado um algoritmo que simula o comportamento do ECC de um bloco de função dentro do ambiente de

edição, contudo, o simulador carecia de elementos para visualização das mudanças de estados, portanto, o código foi atualizado e mais elementos gráficos foram adicionados.

Diante dessa última atualização, mostrou-se necessário a simulação de uma rede de blocos de função. O comportamento interno de blocos de função compostos é definido por uma rede de instâncias de blocos de função. Portanto, a definição inclui conexões de dados e de eventos que precisam existir entre as instâncias internas de blocos de função [4]. Com isso, a aplicação foi expandida com o objetivo de abranger a simulação de uma rede de blocos de função compostos.

No projeto de pesquisa que está sendo continuado, estava definido como trabalhos futuros a integração do FBE 3 com um software para modelagem de autômatos, como o Nadzoru [7], ambiente de criação, edição e exportação de autômatos de estado finito, desenvolvido pelo GASR. O algoritmo de tradução de um autômato descrito em um arquivo XML para um bloco de função equivalente foi desenvolvido por outro bolsista do projeto de pesquisa [8], entretanto, o algoritmo não possuía integração com o FBE, logo, mostrou-se necessária a implementação de uma interface de importação dos arquivos XML gerados pelo Nadzoru, ou outro software de modelagem de autômatos.

RESULTADOS

O FBE passou a simular as máquinas de estado dos blocos criados e importados, através de elementos gráficos provenientes do *toolkit* PyQt 6, mostrando todos os estados e transições do ECC daquele bloco, bem como seu funcionamento. A cor do estado do autômato em que a simulação se encontra muda para uma cor diferente da original, evidenciando para o usuário sua sequência lógica. Além disso, uma aba lateral exibe o estado naquele instante de todos os eventos que constam no autômato (Figura 1).

Além da simulação de um bloco singular, a aplicação agora simula uma rede de blocos de função compostos, exibindo graficamente os eventos, as variáveis e seus respectivos valores no bloco. Dessa forma, o usuário define o valor das variáveis de entrada, e quando executado, o algoritmo retorna com elementos visuais do estado das variáveis de saída em um *log* de execução (Figura 2).

Por fim, depois de implementada a interface que importa para o FBE um arquivo XML definindo um autômato, o usuário passou a poder visualizar no FBE o bloco de função construído de acordo com a definição formal do autômato finito, através de uma representação gráfica explícita, podendo então editá-lo no ambiente e fazer o download do bloco localmente.

CONSIDERAÇÕES FINAIS

Considerando os resultados obtidos, o projeto agora dispõe de ferramentas e funcionalidades essenciais para um funcionamento de edição, simulação, execução, importação e exportação de sistemas de blocos de função, bem como a interface de importação e tradução de um autômato para FB. Como trabalhos futuros, propõe-se o desenvolvimento e a integração de uma interface de comunicação com dispositivos externos ao FBE 3, como microcontroladores, placas de prototipagem como Arduino, CLPs, visando também a aplicação direta desse trabalho em robôs móveis e-puck [9]. O objetivo é permitir que a ferramenta estabeleça comunicação via rede com outros dispositivos, traduzindo a rede de FB gráfica em comandos de configuração para serem enviados e interpretados por ambientes de execução aderentes à norma IEC 61499.

Palavras-chave: function block environment; IEC 61499; bloco de função; simulação.

ANEXOS

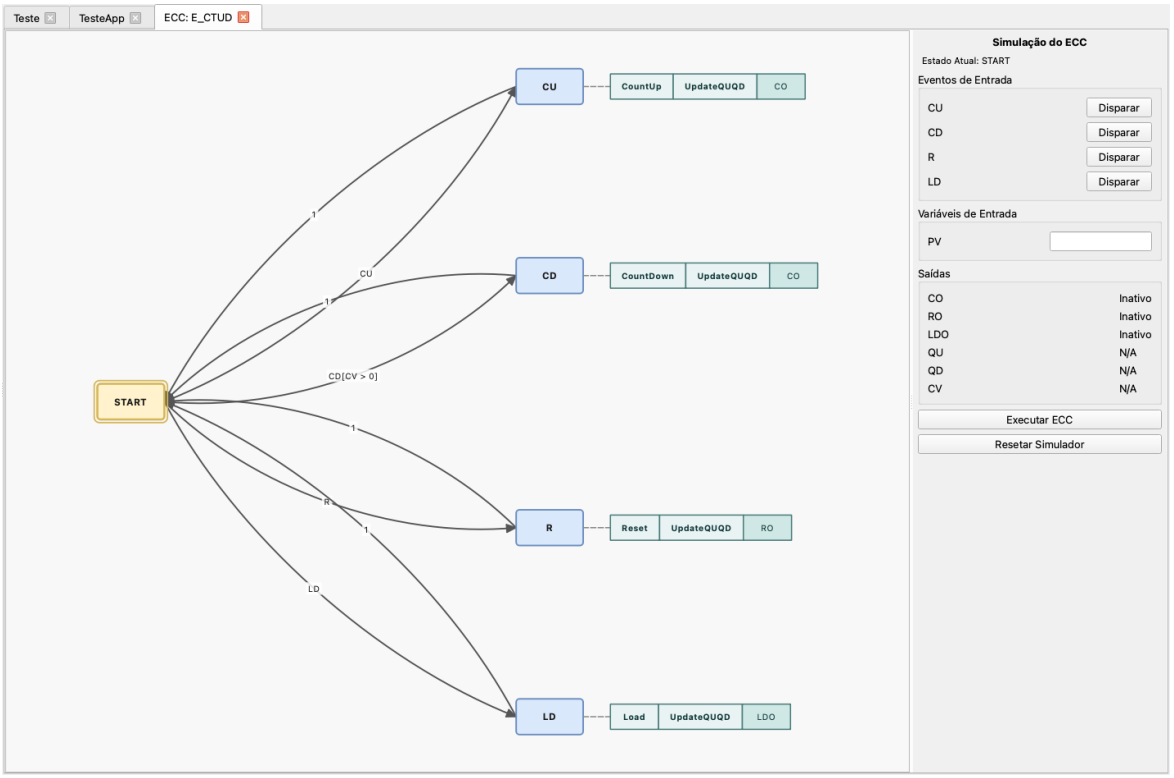


Figura 1. *Simulador do ECC de um bloco de função.*

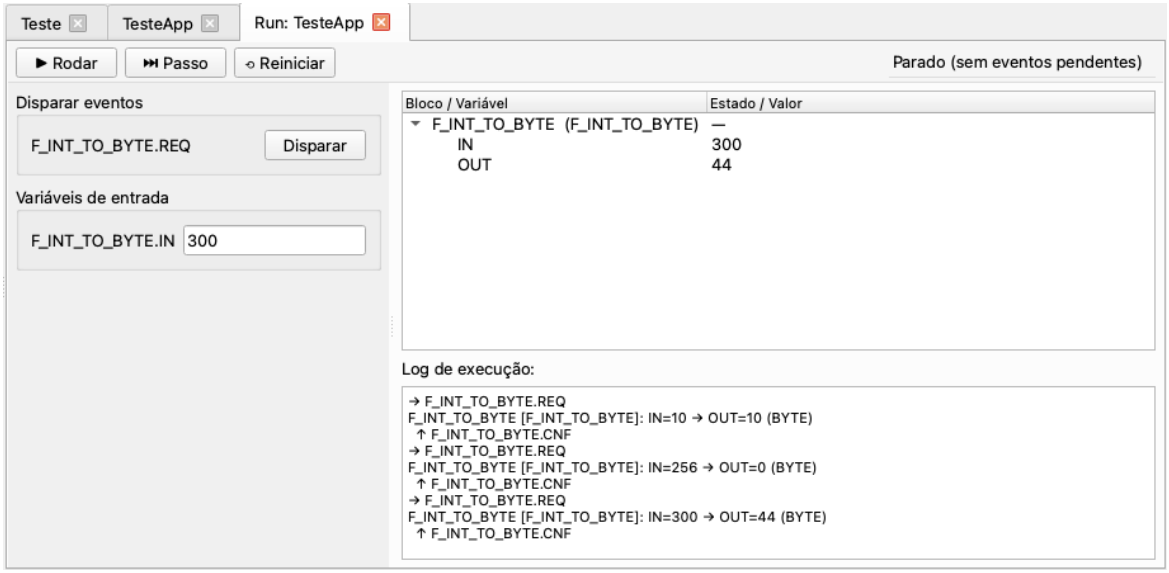


Figura 2. *Simulador da rede de bloco de função composto.*

REFERÊNCIAS BIBLIOGRÁFICAS

[1] IEC61499-1. **Function blocks - Part 1: Architecture**. Geneva: International Electrotechnical Commission, nov. 2012.

[2] IEC61499-2. **Function blocks - Part 2: Software and tool requirements**. Geneva: International Electrotechnical Commission, nov. 2012.

[3] IEC61499-4. **Function blocks - Part 4: Rules for compliance profiles**. Geneva: International Electrotechnical Commission, jan. 2013.

[4] ZOITL, Alois; LEWIS, Robert. **Modelling control systems using IEC 61499**. 2nd Edition. [S.l.: s.n.], mai. 2014. ISBN 978-1-84919-760-1. DOI: 10.1049/PBCE095E.

[5] TEAM, The GTK. **The GTK Project - A free and open-source cross-platform widget toolkit**. Disponível em: <https://www.gtk.org/>. Acesso em: 21 set.. 2026.

[6] LIMITED, Riverbank Computing. **PyQt6: Python bindings for the Qt cross platform application toolkit**. Disponível em: <https://pypi.org/project/PyQt6/>. Acesso em: 21 set.. 2026.

[7] PINHEIRO, Lucas Preischadt et al. Nadzoru: **A Software Tool for Supervisory Control of Discrete Event Systems**. IFAC-PapersOnLine, v. 48, n. 7, p. 182–187, 1 jan. 2015.

[8] FERREIRA, L. Z.; LEAL, A. B.; LEITÃO, H. A. S.; LOPES, Y. K.; ROSSO JUNIOR, R. S. U. **Desenvolvimento de Software para conversão de Sistemas a Eventos Discretos baseados em Autômatos para Blocos de Funções Aderentes a IEC 61499**. In: 35o SIC UDESC, 2025.

[9] MONDADA, F. et al. **The e-puck, a robot designed for education in engineering**. In: Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions. [S.l.: s.n.], 2009. v. 1, n. 1, p. 59–65.