Implementação da Estrutura de Controle Modular Local Aderente à IEC 61499 e Usando OPC-UA

Ricardo Paes Rech * **André Bittencourt Leal** *

* *Programa de Pós-graduação Profissional em Engenharia Elétrica - PPGPEE , Universidade do Estado de Santa Catarina, SC, (e-mail:ricardorech@hotmail.com, andre.leal@udesc.br)*

Abstract: This work presents a methodology for implementing the Local Modular Control Structure in Function Blocks of the IEC 61499 standard, in which the OPC-UA protocol is used in the communication between the plant and the controller. The presentation of the methodology is based on a case study, developed on a virtual plant of the FACTORY I/O, and the control of this plant is done through the FORTE execution environment, of the Eclipse 4diac platform.

Resumo: Este trabalho apresenta uma metodologia para implementação da estrutura de controle Modular Local em Blocos de Função da norma IEC 61499 e na qual é usado o protocolo OPC-UA na comunicação entre a planta e o controlador. A apresentação da metodologia é feita sobre um estudo de caso, desenvolvido sobre uma planta virtual do FACTORY I/O, e o controle dessa planta é feito por intermédio do ambiente de execução FORTE, da plataforma Eclipse 4diac.

Keywords: Discrete Event Systems, Local Modular Approach, IEC 61499, OPC-UA, Function Blocks, FACTORY I/O, 4diac.

Palavras-chaves: Sistemas a Eventos Discretos, Abordagem Modular Local, IEC 61499, OPC-UA, Blocos de Função, FACTORY I/O, 4diac.

1. INTRODUÇÃO

Introduzida por Queiroz e Cury (2002), a abordagem modular local de síntese de supervisores para Sistemas a Eventos Discretos (SEDs) consiste em uma técnica baseada em métodos formais para a obtenção de supervisores locais, os quais, ao atuarem em conjunto, asseguram uma lógica de controle que atende às especificações de controle de forma minimamente restritiva e não bloqueante. Muitos trabalhos sobre a implementação da estrutura de controle modular local são encontrados na literatura, mas praticamente todos eles se baseiam na IEC 61131, dentre os quais se podem citar os trabalhos de Leal et al. (2009) e Vieira et al. (2017). Uma visão geral das abordagens formais para a síntese e implementação em controladores lógicos aderente à IEC61131 é apresenta por Zaytoon e Riera (2017).

A norma IEC 61499 surgiu como uma evolução da IEC 61131 e, dentre outras vantagens, oferece suporte para portabilidade, interoperabilidade e reconfiguração, cobrindo lacunas importantes deixadas pela IEC 61131 e indo ao encontro das necessidades da indústria 4.0 e sistemas ciberfísicos (CPS), conforme apontam Lyu e Brennan (2021) e Vyatkin (2011). O principal componente da IEC 61499 são os Function Blocks (FBs), que contêm os algoritmos para execução das tarefas de controle. Deve-se ressaltar que os FBs consistem numa evolução dos blocos de função da norma IEC 61131. A IEC 61499 adota os eventos como forma de controlar o fluxo de execução dos FBs, enquanto a IEC 61131 faz uma execução sequencial de todos os blocos. Como a execução dos FBs se dá pela ocorrência de eventos, de acordo com os *Execution Control Charts* (ECCs), que são modelados por máquinas de estados, existe forte relação entre essa norma e a teoria de controle supervisório (TCS) de SEDs. Abordagens para migrar soluções desenvolvidas em IEC 61131 para IEC 61499 são discutidas em Dai e Vyatkin (2012) além de apresentar um estudo de caso para um carrossel de bagagem em aeroporto. Exemplos de aplicações avançadas da IEC 61499 tais como detecção de falhas e reconfiguração dinâmica são apresentados em Leitão et al. (2020) e Pinto et al. (2017), respectivamente. Em ambos os caso os autores fazem uso da TCS para obtenção de uma solução ótima e não bloqueante, mas a implementação é feita a partir do autômato do supervisor monolítico. Terzimehic et al. (2017) também abordam em seu trabalho o conceito de reconfiguração dinâmica baseado no conceito de serviços de reconfiguração que a IEC 61499 apresenta e orquestração de serviços via OPC-UA, porém sem empregar métodos formais. A IEC 61499 vem sendo pesquisada tanto para soluções na indústria de batelada, conforme apresentado por Thramboulidis et al. (2007), como para processos contínuos (García et al. (2017)) e indústria de manufatura (Ye e Hong (2018)), que será foco deste artigo, sendo que os dois últimos trabalhos utilizam o protocolo OPC-UA para integração entre dispositivos. Salazar e Alvarado (2014) apresentam um paralelo entre a IEC 61131 e a IEC 61499 contrapondo os paradigmas de programação, porém sem apresentar aspectos relacionados à implementação. Salazar e Alvarado (2014) apontam ainda algumas ferramentas aderentes à IEC 61499, tal como FDBK, ISaGRAF e OOONEIDA Workbench. Outra ferramenta aderente à norma apresentada por Strasser et al. (2008) é o *software* Eclipse 4diac (https://www.eclipse.org/4diac/), que será

adotado para o desenvolvimento deste trabalho, pois além do ambiente de desenvolvimento integrado (IDE), conta também com um ambiente de execução (RTE) que permite embarcar a solução em dispositivos programáveis. A apresentação de um CPS para indústria é abordada por Dai et al. (2018), na qual uma das principais características é a transparência de informação viabilizando a interoperabilidade entre diversos dispositivos, utilizando para tal os FBs da IEC 61499 e o protocolo OPC-UA, porém sem usar métodos formais na solução.

Propostas de implementação da estrutura de controle supervisório modular local aderente à IEC 61499 são apresentadas por Kühl (2018) e Foyth (2019), porém apresentando os resultados em forma de simulação e sem o uso de protocolos de comunicação. Já Mendonça (2019) faz melhorias sobre a proposta anterior e introduz a DES4DC (Discrete Event System for Distributed Control) também usando os FBs da IEC 61499 e cobrindo temas não abordados por Kühl (2018), como o efeito avalanche. Prado (2019) faz uso da DES4DC agregando a geração automática de código para criar os FBs.

O presente trabalho propõe uma metodologia diferente da DES4DC, criando FBs que representam os supervisores obtidos com a abordagem modular local, porém sem a necessidade de criar FBs para análise de eventos não controláveis e ainda explorando a comunicação em OPC-UA para integrar componentes fora do ambiente de desenvolvimento da IEC 61499, o que viabiliza sua implementação em caráter mais amplo tais como aplicações em CPS. Os trabalhos de Kühl (2018), Foyth (2019), Mendonça (2019) e Prado (2019) foram desenvolvidos no contexto do mesmo grupo de pesquisa que o presente trabalho e sob a supervisão do mesmo orientador. Assim, a presente proposta consiste num aprimoramento das anteriores, que não foram publicadas na forma de artigo. Sendo assim, uma contribuição esperada com esta publicação é a de dar maior visibilidade para o tema, que a despeito de sua relevância, ainda é pouco investigado no Brasil. Para facilitar a compreensão, a metodologia proposta é apresentada a partir de um estudo de caso de uma planta de manufatura virtual simulada através do FACTORY I/O (2022), que atua como OPC-UA Client, lendo e escrevendo dados para o 4diac, que desempenha o papel de OPC-UA Server. Antes de apresentar a metodologia proposta, são apresentadas as ferramentas de desenvolvimento e simulação usadas no trabalho.

2. FERRAMENTAS DE DESENVOLVIMENTO E SIMULAÇÃO

A implementação de soluções de automação industrial em FBs da IEC 61499 requer a utilização de ferramentas que sejam aderentes à norma. Estas ferramentas já trazem consigo uma vasta biblioteca com FBs de uso geral. No entanto, para a implementação usando a metodologia proposta é necessária a criação de FBs que atendam o comportamento dos autômatos obtidos na resolução do problema de controle supervisório, devendo-se, portanto, compilar os novos FBs. Esta seção apresenta as ferramentas usadas no processo de implementação da estrutura de controle obtida com a adoção da metodologia proposta neste trabalho.

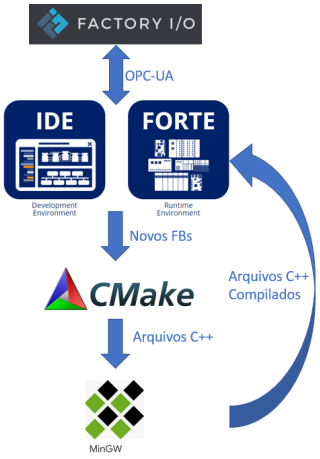

Figura 1. Integração das Ferramentas

O **4diac IDE** é um ambiente integrado de desenvolvimento *open-source* que permite criar modelos de controle distribuído aderentes à IEC 61499, bem como FBs, aplicativos e configuração de dispositivos entre outras tarefas relacionadas à norma.

O **4diac FORTE** é um RTE (*Run Time Environment*) baseado em C++, que suporta a execução de FBs aderentes a IEC 61499 em dispositivos embarcados. Pode ser executado em sistemas operacionais e até mesmo RTOS (*Real Time Operating Systems*) pois trata-se de um ambiente de execução *multi-threaded*, com baixo consumo de memória.

CMake é um conjunto de ferramentas multiplataforma gratuito e *open-source* para teste, automatização, empacotamento e geração de executáveis que usa um método independente de compilador. O CMake pode gerar arquivos de projeto para diversas IDEs sendo uma delas o Eclipse, que é usado como base para o 4diac.

MinGW é um acrônimo para Minimalist GNU for Windows. Permite que desenvolvedores no Windows utilizem o GCC (GNU Compiler Collection) sem precisar emular um sistema Unix-like.

O **FACTORY I/O** é um ambiente para simulação e desenvolvimento de sistemas de automação industrial que permite criar plantas virtuais com sensores e atuadores que podem ser lidos e controlados por dispositivos externos, tais como CLPs ou microcontroladores. A integração entre o FACTORY I/O e os dispositivos externos é feito por protocolos de comunicação, tal como Modbus-TCP ou UPC-UA.

OPC-UA é um protocolo de comunicação industrial multiplataforma, que possibilita a troca de dados entre diversos dispositivos. Como ele não é um protocolo nativo no 4diac, faz-se necessário adicionar e compilar uma biblioteca à parte. Escolhemos a open62541, uma implementação de OPC-UA gratuita, independente de plataforma e de código aberto. A biblioteca fornece as ferramentas necessárias para implementar OPC-UA *clients* e *serves* dedicados, ou para integrar a comunicação baseada em OPC-UA em aplicativos existentes.

Vale destacar que para o cálculo dos supervisores modulares locais foi usada a ferramenta Nadzoru (Pinheiro et al. (2015)).

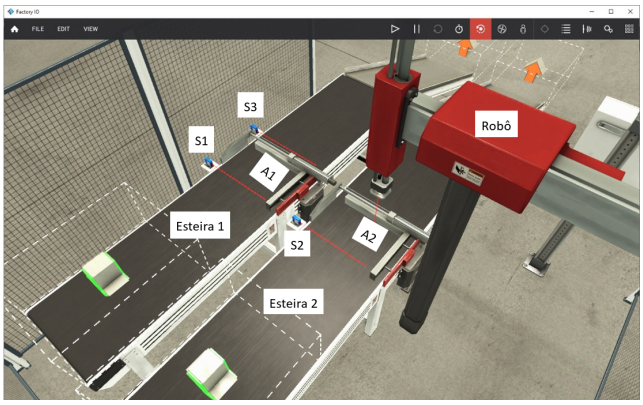

Figura 2. Sistema Assembler do Factory I/O

2.1 Integração das Ferramentas

O 4diac IDE é usado no desenvolvimento dos novos FBs, e na interligação dos FBs que compõem a solução completa, além de ser usado no debug dos FBs, ao passo que o 4diac FORTE executa os FBs. O CMake é usado para gerar os arquivos em C++ referente aos FBs criados no 4diac, enquanto o MinGW é responsável pela compilação dos arquivos gerados pelo CMake. O FACTORY I/O simula uma planta fabril e se comunica com o 4diac FORTE via OPC-UA. O diagrama que representa esta estrutura é apresentado na Figura 1.

3. APRESENTAÇÃO DA METODOLOGIA A PARTIR DE UM ESTUDO DE CASO

3.1 Planta Industrial no FACTORY I/O

O FACTORY I/O vem com diversas plantas industriais pré-configuradas, chamadas *scenes*. Para o estudo de caso, escolhemos arbitrariamente a planta *Assembler*, cujo objetivo consiste em fazer a montagem de uma tampa sobre uma base. Este sistema é composto por duas esteiras transportadoras (E1 e E2), três sensores de posição (S1, S2 e S3), dois atuadores (A1 e A2) e um braço robótico (R), conforme ilustrado na Figura 2. As bases entram pela esteira E1 e as tampas pela esteira E2 e a montagem é feita pelo Robô, que retira uma tampa de E2 e a coloca sobre uma base em E1.

A esteira E2 deve ser desligada quando uma tampa alcança a posição de pega, o que se dá pela ocorrência da desativação de S2. O atuador A1 (A2) deve ser fechado para garantir que a base (tampa) esteja na posição correta para montagem (pega). Após uma peça ser montada, A1 deve ser elevado para que ela avance pela E1, deixando a estação, quando então A1 deve ser baixado.

3.2 Solução do Problema de Controle Supervisório

Nesta seção apresenta-se a solução para o problema de controle supervisório definido sobre a planta *Assembler*, solução essa obtida por intermédio do uso da abordagem modular local de síntese de supervisores. Na Figura 3 são apresentados os autômatos usados para representar o comportamento dos subsistemas que compõem a planta, chamados de sistema produto (SP1 à SP8). Os eventos

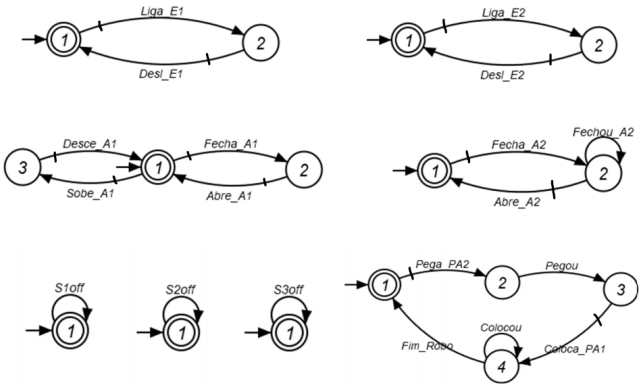

Figura 3. Autômatos para os subsistemas (Esteiras E1, E2; Atuadores A1, A2; Sensores S1, S2, S3; Robô R.)

controláveis são denotados por um traço cortando as setas que representam as transições, como é o caso, por exemplo, dos eventos Liga_E1 e Desl_E1 no modelo SP1, que representam comandos do CLP para ligar e desligar a esteira E1. Vale destacar que na modelagem dos sensores não foram incluídos os eventos de ativação (identificação de presença de peça), pois eles não são relevantes para a solução do problema. Além disso, adotou-se um certo grau de abstração na modelagem do robô visando a obtenção de um modelo razoavelmente pequeno, mas que permite a operação do sistema com eficiência, permitindo, por exemplo, que o robô pegue uma tampa mesmo quando ainda não chegou uma base para a montagem. Na Tabela 1 é apresentada uma breve descrição dos eventos controláveis (C) e não controláveis (NC) usados na modelagem da planta.

Tabela 1. Descrição dos eventos usados na modelagem da planta

Evento	C/NC	Descrição
Liga_E1	C	Comando para ligar a esteira E1
Desl_E1	C	Comando para desligar a esteira E1
Liga_E2	C	Comando para ligar a esteira E2
Desl_E2	C	Comando para desligar a esteira E2
Desce_A1	C	Comando para descer o atuador A1
Sobe_A1	C	Comando para subir o atuador A1
Fecha_A1	C	Comando para fechar o atuador A1
Abre_A1	C	Comando para abrir o atuador A1
Fecha_A2	C	Comando para fechar o atuador A2
Abre_A2	C	Comando para abrir o atuador A2
Pega_PA2	C	Comando para R pegar tampa na E2
Coloca_PA1	C	Comando para R colocar tampa sobre base
S1off	NC	Sinal de desativação do sensor S1
S2off	NC	Sinal de desativação do sensor S2
S3off	NC	Sinal de desativação do sensor S3
Fechou_A2	NC	Sinal de sensor indicando que A2 fechou
Pegou	NC	R pegou uma tampa na esteira E2
Colocou	NC	R colocou tampa sobre a base em E1
Fim_Robo	NC	Sinal de fim de operação do robô

Os modelos para as especificações de controle (E1 à E12), que correspondem também aos respectivos supervisores reduzidos (SR1 à SR12), são mostrados na Figura 4. Pode-se resumir as especificações de controle conforme segue:

- Fechar o atuador A1 (A2) somente quando o sensor S1 (S2) é desativado.
- Desligar a esteira E1 (E2) sempre que ocorre o fechamento de A1 (A2).

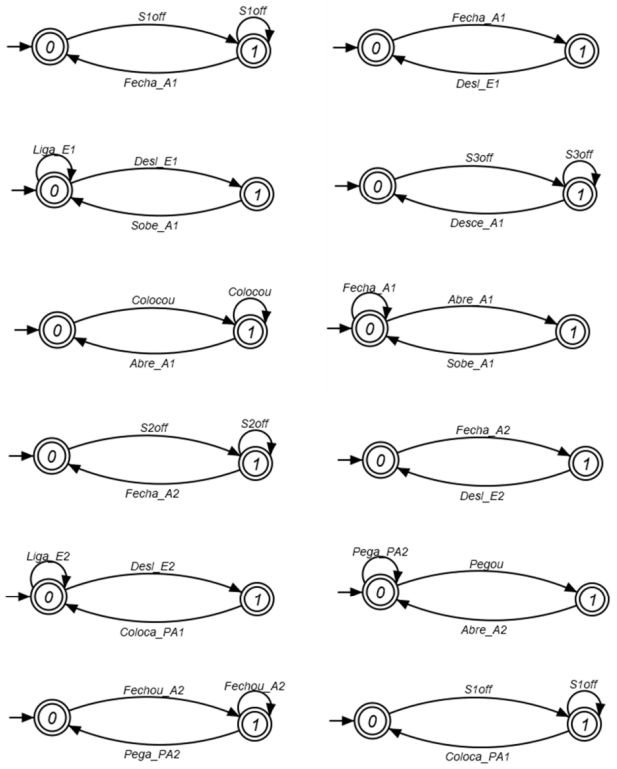

Figura 4. Autômatos para os Supervisores Reduzidos (SR1 à SR12)

- Se E1 é desligada, ela só deve ser ligada novamente após ocorrer a subida de A1, com a liberação da peça montada.
- A1 só deve descer para interceptar nova base após haver a desativação de S3.
- A1 só deve abrir após o robô ter colocado uma tampa sobre a base (robô informar o evento Colocou).
- A1 só pode subir após abrir e, uma vez aberto, só pode fechar após subir.
- E2 deve ser desligada para que R possa transportar a tampa (comando Coloca_PA1) e só pode ser religada após o início deste transporte.
- A2 só pode abrir após R ter pego a tampa.
- R só pode pegar tampa na E2 se A2 fechou.
- R só pode colocar tampa sobre base (comando Coloca_PA1) se houver base no local de montagem (após S1off).

3.3 Metodologia de Implementação Proposta

Na Figura 5 ilustra-se a estrutura que representa a metodologia de implementação proposta, em que os retângulos em azul representam FBs, o retângulo em cinza é a planta a ser controlada, e as setas indicam o fluxo de dados. A metodologia prevê a criação de FBs que representam os subsistemas (SP) e os supervisores reduzidos (SR). É necessário criar blocos de análise responsáveis pelo sincronismo dos eventos controláveis (EC) e FBs de sequência operacional (SO) que fazem a interface entre a solução teórica e o sistema físico a ser controlado. Os demais blocos usados são nativos do 4diac. A interligação entre os conjuntos de FBs, bem como a explicação sobre cada um

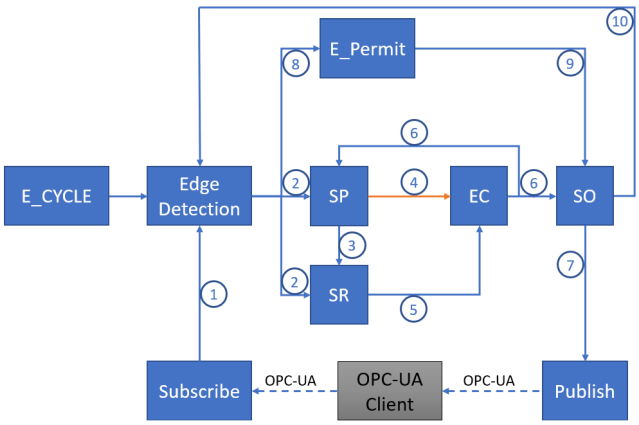

Figura 5. Estrutura da Metodologia Proposta

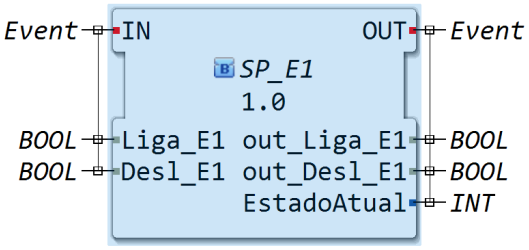

Figura 6. FB SP1 Esteira E1 - Interface

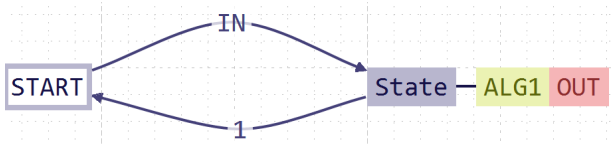

Figura 7. FB SP1 Esteira E1 - ECC

é dada nas seções subsequentes, juntamente com o estudo de caso.

3.4 Aplicação da metodologia ao estudo de caso

Para o sistema Assembler, deve-se criar FBs para os autômatos que representam os SPs das esteiras E1 e E2, atuadores A1 e A2, e para o Robô. Não há necessidade de implementar nenhum algoritmo para representar os sensores S1, S2 e S3, uma vez que nesses modelos se considera apenas a desativação do sensor e isso pode ser alcançado com um bloco nativo do 4diac, como mostraremos mais adiante. Para a esteira E1, por exemplo, a interface do FB é apesentada na Figura 6, seu ECC na Figura 7, e o algoritmo que descreve o funcionamento do autômato na Figura 8.

Para cada um dos SRs foi criado um FB. Nas Figuras 9, 10, 11 e 12 ão apresentados, respectivamente, o FB, seu ECC e dois algoritmos para o SR1.

Seguindo a metodologia proposta, foram criados também blocos de análise de EC. O objetivo deste FB é garantir que um evento controlável só será gerado caso ele tenha sido habilitado por todos os supervisores que tratam deste evento e que ele esteja ativo no estado atual do SP a ele relacionado. Para ilustrar, toma-se como base o FB EC_Abre_A1, ilustrado na Figura 13. Como o evento Abre_A1 faz parte do alfabeto de SR5 e SR6, para que

Algorithm ALG1

Language ST Comment New Algorithm

```
1  //Inicializa estados, pois variáveis internas não estão assumindo
2  //o valor atribuido em "Initial Value"
3  if E1 then
4     E2 := false;
5  end_if;
6
7  if E2 and Desl_E1 then
8  E1 := true;
9  E2 := false;
10 Desl_E1 := false;
11 out_Desl_E1 := true;
12 else
13 out_Desl_E1 := false;
14 end_if;
15
16 if E1 and Liga_E1 then
17 E2 := true;
18 E1 := false;
19 Liga_E1 := false;
20 out_Liga_E1 := true;
21 else
22 out_Liga_E1 := false;
23 end_if;
24
25 if E1 then
26 EstadoAtual := 1;
27 else
28 EstadoAtual := 2;
29 end_if;
```

Figura 8. FB SP1 Esteira E1 - Algoritmo

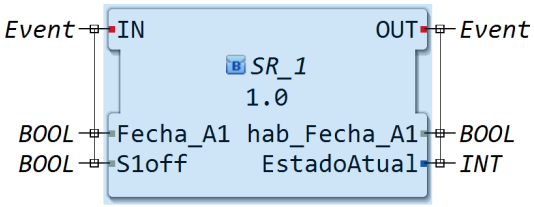

Figura 9. FB SR1 - Interface

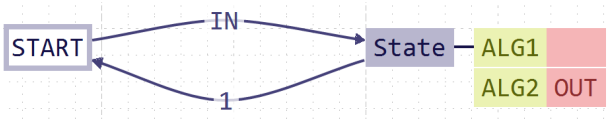

Figura 10. FB SR1 - ECC

ele seja gerado é necessário que estes dois supervisores habilitem a sua geração (sinais Abre_A1_1 e Abre_A1_2). Além disso, para que o evento possa ocorrer na planta é necessário que o SP_A1 esteja no estado 2. Nas Figuras 14 e 15 são apresentados o ECC e o algoritmo para este FB.

Outro conjunto de FBs é chamado de Sequência Operacional (SO), responsável por adequar o nível de abstração usado na modelagem ao comportamento efetivo da planta. O melhor exemplo para ilustrar este FB é o caso do robô, que conta com sensores detecção de peça e de movimentação no eixo X e no eixo Z. O robô recebe os comandos para mover no eixo X, no eixo Z e para pegar peça. Estes sinais diferem dos eventos considerados na modelagem SP do robô. Assim o FB SO_Robo apresentado na Figura 16 faz a adequação entre os eventos considerados na modelagem do

Algorithm ALG1

Language ST Comment New Algorithm

```
1  //Inicializa estados, pois variáveis internas não estão assumindo
2  //o valor atribuido em "Initial Value"
3  if E1 then
4     E2 := false;
5  end_if;
6
7  if E1 and S1off then
8  E1 := false;
9  E2 := true;
10 S1off := false;
11 end_if;
12
13 if E2 and S1off then
14 S1off := false;
15 end_if;
16
17 if E2 and Fecha_A1 then
18 E1 := true;
19 E2 := false;
20 Fecha_A1 := false;
21 end_if;
```

Figura 11. FB SR1 - ALG1

Algorithm ALG2

Language ST Comment New Algorithm

```
1  if E2 then
2  hab_Fecha_A1 := true;
3  EstadoAtual := 2;
4  end_if;
5
6  if E1 then
7  hab_Fecha_A1 := false;
8  EstadoAtual := 1;
9  end_if;
```

Figura 12. FB SR1 - ALG2

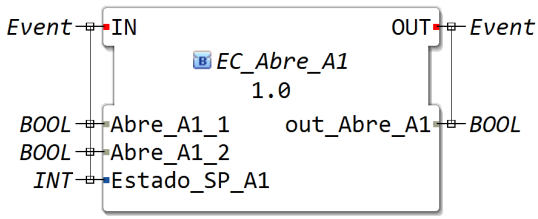

Figura 13. FB Evento Controlável Abre A1 - Interface

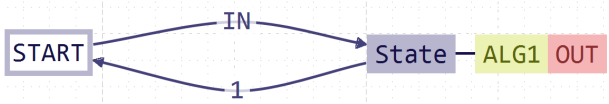

Figura 14. FB Evento Controlável Abre A1 - ECC

robô e os sinais elétricos dos sensores e atuadores que que fazem parte do robô. O ECC e o primeiro algoritmo deste bloco estão nas Figuras 17 e 18. Por limitação de espaço não apresentamos todos os algoritmos usados na solução.

3.5 FBs adicionais necessários para solução do estudo de caso

Além dos FBs desenvolvidos foram usados alguns blocos nativos do 4diac, a saber: **E_CYCLE**, responsável por atu-

Algorithm ALG1

Language ST Comment

```
1  if Estado_SP_A1 = 2 and (Abre_A1_1 and Abre_A1_2) then
2     out_Abre_A1 := true;
3  else
4     out_Abre_A1 := false;
5  end_if;
```

Figura 15. FB Evento Controlável Abre A1 - Algoritmo

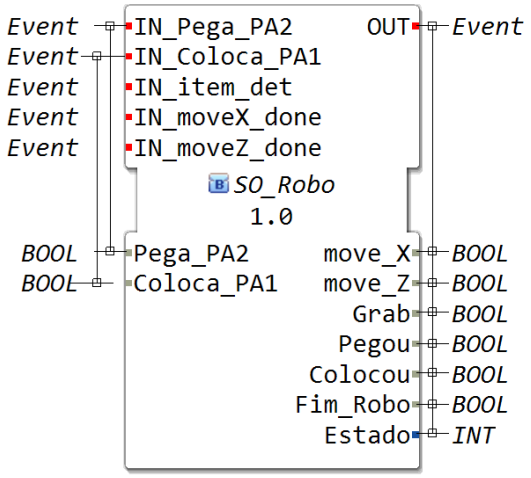

Figura 16. FB Sequência Operacional Robô - Interface

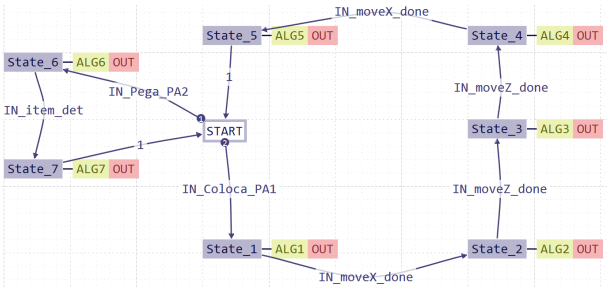

Figura 17. FB Sequência Operacional Robô - ECC

Algorithm ALG1

Language ST Comment

```
1  if Coloca_PA1 then
2     Fim_Robo := false;
3     move_X := true;
4     Estado := 1;
5  end_if;
6
```

Figura 18. FB Sequência Operacional Robô - ALG1

alizar o sistema dando um pulso a cada 100ms; **R_TRIG** e **F_TRIG**, que detectam respectivamente borda de subida e borda de descida; **FBs PUBLISH_1** e **SUBSCRIBE_1**, responsáveis por publicar e ler os dados em OPC-UA; **E_PERMIT**, que permite a propagação de um evento.

3.6 Interligação dos FBs

Para que os FBs apresentados nas sessões anteriores executem a lei de controle obtida no projeto é feita a interligação entre os blocos, tal como apresentado na Figura 5 e detalhada a seguir. Note que os blocos na cor azul estão no ambiente do 4diac, enquanto que o bloco em cinza representa o FACTORY I/O.

① Os sinais de saída da planta no FACTORY I/O, referentes aos eventos não controláveis, são lidos em OPC-UA pelos blocos SUBSCRIBE_1 e, usando blocos de detecção de borda Edge Detection, é feita a geração dos respectivos eventos não controláveis. Isso justifica a não criação de FBs para os sensores S1, S2 e S3, uma vez que os blocos F_TRIG cumprem a função de identificar as bordas de descida relativas às suas desativações. Sinais cíclicos do E_CYCLE garantem a leitura periódica das entradas, funcionando como o ciclo de scan de um CLP.

② Os eventos não controláveis que fazem parte dos modelos dos subsistemas são recebidos nos FBs do SP ou do SR. Os eventos que se encontram como selfloops nos modelos dos SPs são ligados diretamente nos SRs, sem passar pelos SPs, uma vez que não é preciso fazer atualização de estados nesses modelos. Os demais são recebidos nos FBs do SP a fim de provocar a evolução de estados dos seus ECCs e depois seguem pelo caminho 3 para atualização de estado dos SRs.

③ Tanto eventos controláveis quanto não controláveis, que precisam ser repassados para o SR a fim de fazer as devidas atualizações de estados nesses FBs.

④ A informação do estado atual existente na saída dos FBs dos SPs alimenta os FBs de ECs e é usada no algoritmo do EC para impedir que ocorra o acionamento de um evento controlável sem que o estado do SP dê condição.

⑤ Os FBs de ECs recebem ainda dados de habilitação provenientes dos blocos de SR para permitir a geração de eventos controláveis.

⑥ Os dados relativos aos eventos controláveis habilitados pelos FBs ECs são repassados aos FBs SO, dando início às respectivas sequências operacionais. Esses mesmos dados são repassados para os FB SP a fim de gerar os respectivos eventos controláveis nestes blocos e fazer a atualização de estados dos seus ECCs.

⑦ Os dados de saída dos FBs de SO, relacionados aos eventos controláveis, modelados ou não nos subsistemas, são publicados em OPC-UA pelos FBs PUBLISH_1 e lidos pelo FACTORY I/O que irá promover as correspondentes atuações na planta.

⑧ Eventos não controláveis que não foram usados na modelagem dos autômatos devido a abstração adotada, mas existem na planta e são usados no controle dos FBs de SO. Os dados lidos dos sensores são convertidos em eventos pelo FB E_PERMIT.

⑨ Eventos não controláveis usados na lógica dos FBs de SO. Por exemplo, na Figura 17 os eventos IN_item_det, IN_moveX_done, IN_moveZ_done existem na planta, mas não foram modelados nos autômatos, tal como os eventos IN_pega_PA1 e IN_coloca_PA1.

⑩ Eventos não controláveis que foram usados na solução teórica, mas que não existem na planta; realimentam os FBs de SPs, passando primeiro pelos blocos de detecção de borda.

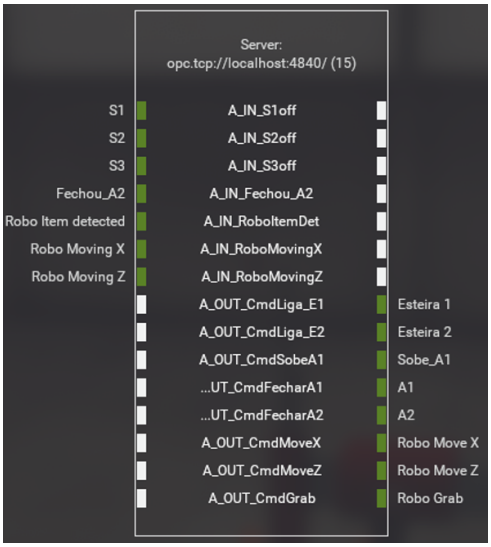

Figura 19. Configuração OPC-UA no FACTORY I/O

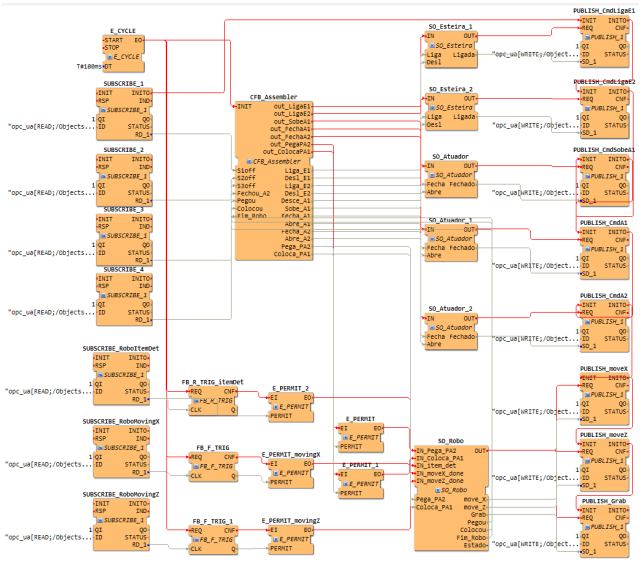

Figura 20. Solução completa - 4diac

A Figura 19 apresenta os dados trocados entre o 4diac e o FACTORY I/O, sendo que à esquerda do bloco estão dispostos os sensores e à direita os atuadores.

3.7 Apresentação da solução completa

A solução completa, com a interligação entre todos os FBs é apresentada na Figura 20. Observar que para efeito de organização foi criado o CFB_Assembler, um bloco composto que embarca a lei de controle da planta dentro de um único componente. Na Figura 21 é apresentada a interface deste FB, e a Figura 22 exibe os blocos internos, encapsulados pelo CFB_Assembler.

4. CONCLUSÃO

Neste trabalho foi apresentada uma metodologia de implementação da estrutura de controle modular local que, por ser aderente à IEC 61499, herda as boas características dessa norma, quais sejam: a portabilidade, interoperabilidade e reconfiguração dinâmica. Vários testes foram

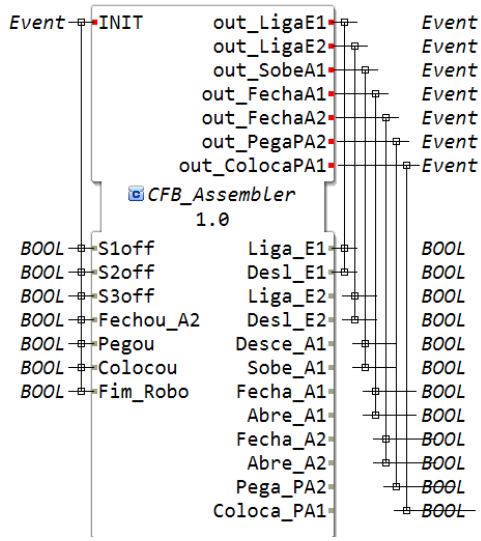

Figura 21. Bloco composto que embarca solução teórica

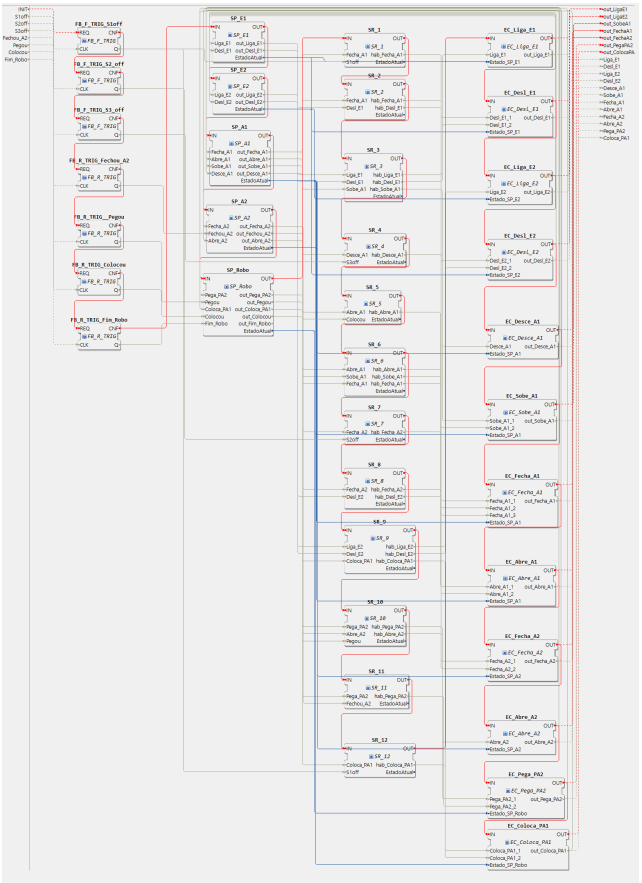

Figura 22. FBs internos ao CFB Assembler

executados e o sistema apresentou os comportamentos esperados. Ressalta-se que muito embora a IEC 61499 seja voltada para a automação distribuída, e que neste trabalho se tenha usado uma estrutura de controle modular, a lógica de controle foi implementada de forma centralizada. Assim, como trabalho futuro, pretende-se adaptar esta metodologia para que a estrutura de controle possa ser facilmente distribuída em mais de um elemento de controle, possivelmente de tecnologias e marcas diferentes, o que poderá ser alcançado através do protocolo OPC-UA. Deve-

se investigar ainda formas alternativas ao FB E_CYCLE, uma vez que a IEC 61499 é orientada a eventos, o uso de uma rotina cíclica para atualização do sistema pode representar mau uso do recurso computacional, pois não há necessidade de atualizar os FBs se não houve mudança no estado da planta. Por fim, destaca-se que muito embora se tenha usado um problema de controle simples, desenvolvido sobre uma planta virtual, a metodologia proposta pode ser aplicada para o controle de plantas industriais, uma vez que a metodologia proposta pode ser aplicada para sistemas maiores e mais complexos, além de fazer uso de OPC-UA, um protocolo amplamente adotado por fabricantes de CLPs e sistemas SCADA.

AGRADECIMENTOS

Os autores agradecem o apoio financeiro parcial da Fundação de Amparo à Pesquisa e Inovação do Estado de Santa Catarina - FAPESC 2021TR930.

REFERÊNCIAS

Dai, W., Song, Y., Zhang, Z., Wang, P., Pang, C., e Vyatkin, V. (2018). Modelling industrial cyber-physical systems using IEC 61499 and OPC UA. In *2018 IEEE 16th Int. Conf. on Industrial Informatics*, 772–777.

Dai, W. e Vyatkin, V. (2012). Redesign distributed PLC control systems using IEC 61499 Function Blocks. *IEEE Trans. on Automation Science and Eng.*, 9(2), 390–401.

FACTORY I/O (2022). FACTORY I/O assembler - assemble parts made of lids and bases using a two-axis pick and place. `https://docs.factoryio.com/manual/scenes/assembler/`. URL `https://docs.factoryio.com/manual/scenes/assembler/`. Acessado em: 16 de abril de 2022.

Foyth, J.O. (2019). *Metodologia De Implementação De Código Aderente À Norma Iec 61499 Baseada Na Teoria De Controle Supervisório De Sistemas A Eventos Discretos*. TCC (Graduação), UDESC. Eng. Elétrica., Joinville.

García, M.V., Irisarri, E., Pérez, F., Estévez, E., e Marcos, M. (2017). An open CPPS automation architecture based on IEC-61499 over OPC-UA for flexible manufacturing in Oil&Gas industry. *IFAC-PapersOnLine*, 50(1), 1231–1238.

Kühl, R.A. (2018). *Metodologia de implementação do controle supervisório modular local aderente à norma IEC 61499*. Dissertação (mestrado), UDESC. Programa de Pós-Graduação em Engenharia Elétrica, Joinville.

Leal, A.B., da Cruz, D.L.L., e Hounsell, M.d.S. (2009). Supervisory control implementation into programmable logic controllers. In *2009 IEEE Conference on Emerging Technologies & Factory Automation*, 1–7.

Leitão, H.A.S., Rosso, R.S.U., Leal, A.B., e Zoitl, A. (2020). Fault handling in discrete event systems applied to IEC 61499. In *25th IEEE Int. Conf. on Emerging Technologies and Factory Automation*, 1039–1042.

Lyu, G. e Brennan, R.W. (2021). Towards IEC 61499-based distributed intelligent automation: A literature review. *IEEE Trans. Ind. Informatics*, 17(4), 2295–2306.

Mendonça, R.d. (2019). *DES4DC: uma metodologia para a implementação da estrutura de controle supervisório modular local usando blocos de função da IEC 61499*. TCC (Graduação), UDESC. Eng. Elétrica., Joinville.

Pinheiro, L.P., Lopes, Y.K., Leal, A.B., e Rosso Junior, R.S.U. (2015). Nadzoru: A software tool for supervisory control of discrete event systems. *IFAC-PapersOnLine*, 48(7), 182–187. 5th IFAC International Workshop on Dependable Control of Discrete Systems.

Pinto, L.I., Leal, A.B., e Rosso, R.S.U. (2017). Safe dynamic reconfiguration through supervisory control in IEC 61499 compliant systems. In *IEEE 15th International Conference on Industrial Informatics*, 753–758.

Prado, R.B.S. (2019). *Desenvolvimento e implementação de algoritmos para a conversão automática da estrutura de controle supervisório de SEDs em blocos de função aderentes à IEC 61499 seguindo a metodologia DES4DC*. TCC (Graduação), UDESC. Eng. Elétrica., Joinville.

Queiroz, M.d. e Cury, J. (2002). Synthesis and implementation of local modular supervisory control for a manufacturing cell. In *6th International Workshop on Discrete Event Systems.*, 377–382.

Salazar, L.A.C. e Alvarado, O.A.R. (2014). The future of industrial automation and IEC 61499-3 standard. In *2014 III International Congress of Engineering Mechatronics and Automation (CIIMA)*, 1–5.

Strasser, T., Rooker, M., Ebenhofer, G., Zoitl, A., Sunder, C., Valentini, A., e Martel, A. (2008). Framework for distributed industrial automation and control (4diac). In *6th IEEE Int. Conf. on Ind. Informatics*, 283–288.

Terzimehic, T., Wenger, M., Zoitl, A., Bayha, A., Becker, K., Müller, T., e Schauerte, H. (2017). Towards an industry 4.0 compliant control software architecture using IEC 61499 & OPC UA. In *22nd IEEE Int. Conf. on Emerging Techn. on Factory Autom.*, 1–4.

Thramboulidis, K., Sierla, S., Papakonstantinou, N., e Koskinen, K. (2007). An IEC 61499 based approach for distributed batch process control. In *2007 5th IEEE Int. Conf. on Industrial Informatics*, volume 1, 177–182.

Vieira, A.D., Santos, E.A.P., de Queiroz, M.H., Leal, A.B., de Paula Neto, A.D., e Cury, J.E.R. (2017). A method for plc implementation of supervisory control of discrete event systems. *IEEE Transactions on Control Systems Technology*, 25(1), 175–191.

Vyatkin, V. (2011). IEC 61499 as enabler of distributed and intelligent automation: State-of-the-art review. *IEEE Trans. on Industrial Informatics*, 7(4), 768–781.

Ye, X. e Hong, S.H. (2018). An automationML/OPC UA-based Industry 4.0 solution for a manufacturing system. In *IEEE 23rd Int. Conf. on Emerging Technologies and Factory Automation*.

Zaytoon, J. e Riera, B. (2017). Synthesis and implementation of logic controllers – a review. *Annual Reviews in Control*, 43, 152–168.