

UMA PROPOSTA DE APLICAÇÃO DA MBSE E FPGA PARA CONTROLE DE MICRORREDES BASEADO NA ARQUITETURA IEC 61850

MIGUEL ANGEL ORELLANA POSTIGO*JOSE REINALDO SILVA†

**Departamento de Engenharia Eletrônica / Elétrica
Escola Superior de Tecnologia - Universidade do Estado do Amazonas.
Av. Darcy Vargas, 1200 Parque Dez, CEP 69050-020
Manaus/AM, Brasil*

*†Design Lab, Departamento de Engenharia Mecatrônica
Escola Politécnica - Universidade de São Paulo
Av. Professor Melo Moraes, 2231, CEP 05508-970
São Paulo/ SP, Brasil*

mpostigo@uea.edu.br, reinaldo@usp.br

Abstract— Microgrids emerge as a practical, sustainable, and reliable solution to meet the demand of users who, for various reasons, have restricted access to electricity. However, the flexibility of microgrid systems requires considerable creative engineering effort, especially in the design process. The requirements analysis has a leading role in combining adaptability to users and good performance, better characterizing, understanding, and specifying the application domain and the problem that the microgrids must solve, then embodying and identifying solutions. This work seeks to contribute to a method (Orellana Postigo, 2021) inserted in the MBSE (Model-Based Systems Engineering) approach to model microgrids using IEC 61850 as a reference architecture for the preliminary design. Requirements would be represented in Objective Oriented Requirements Engineering (GORE), using visual diagrams based on the KAOS (Keep All Objectives Satisfied) method, but formally verified using Petri Nets. A case study shows how a preliminary design with matching goals, sub-goals, and algorithms is modeled in Petri Nets and transferred to Field Programmable Gate Array (FPGA), closing the proposed design cycle.

Keywords— Systems design, formal modeling, Petri nets, microgrid, FPGA

Resumo— As microrredes surgem como uma solução prática, sustentável e confiável para atender a demanda de usuários que, por diversos motivos, possuem acesso restrito à rede elétrica. No entanto, a flexibilidade dos sistemas de microrredes requerem considerável esforço e criatividade da engenharia, especialmente na fase de projeto. A análise de requisitos tem um papel preponderante ao aliar adaptabilidade aos usuários e bom desempenho, caracterizando, compreendendo e especificando melhor o domínio de aplicação e o problema que as microrredes devem resolver, concretizando e identificando soluções. Este trabalho busca contribuir com um método (Orellana Postigo, 2021) inserido na abordagem MBSE (Model-Based Systems Engineering) para modelar microrredes utilizando a norma IEC 61850 como arquitetura de referência para o projeto preliminar. Os requisitos seriam representados em Engenharia de Requisitos Orientada a Objetivos (GORE), usando diagramas visuais baseados no método KAOS (Keep All Objectives Satisfied), verificados formalmente usando redes de Petri. Em um estudo de caso mostramos como um projeto preliminar com correspondência de objetivos, sub-objetivos e algoritmos é modelado em Redes de Petri e transferido para Field Programmable Gate Array (FPGA), fechando o ciclo de projeto proposto.

Palavras-chave— Projeto de sistemas, modelagem formal, Redes de Petri, microrredes, FPGA

1 Introdução

As microrredes inteligentes no Brasil são consideradas essenciais para o atendimento de comunidades isoladas (Falcão, 2010), especialmente na região amazônica. Entretanto, existe uma discussão sobre os padrões técnicos e legais para regular esta operação, tanto em sistemas autônomos como integrados à rede elétrica, levando em consideração a aplicação, disponibilidade local de recursos energéticos, dispersão, nível das cargas dos consumidores, e isolamento geográfico dos centros de controle da concessionária (Silva, 2016). Portanto, é importante estudar as configurações mais apropriadas, que permitam otimizar a utilização dos recursos energéticos, garantindo ao mesmo tempo segurança no suprimento de energia e promovendo a redução tarifária.

A literatura mostra diferentes estudos nessa direção, realizados em diferentes países. No entanto, os avanços obtidos são modestos e ainda não ofereceram contribuições que possam ser diretamente transpostas para a realidade brasileira (Silva, 2016), (Postigo and Silva, 2018). Por exemplo, considerando as particularidades do atendimento de comunidades isoladas da região norte do país como exemplo, a dispersão geográfica, a dificuldade de acesso para o transporte de combustíveis e a disponibilidade de recursos locais intermitentes, o equacionamento das principais questões técnicas e de manutenção tornam-se um desafio, o que favorece a aplicabilidade e confiabilidade de projetos de implantação de micro-redes com fontes intermitentes.

Nos últimos anos, alguns trabalhos propuseram a utilização de um ambiente de modelagem

digital para tratar sistemas complexos, criando uma abordagem baseada em padrões para documentar o sistema, que poderia ser formalmente verificada para remover inconsistências no modelo e gerar um *roadmap* para a harmonização dos requisitos de todas as partes interessadas (Postigo et al., 2020), (Uslar et al., 2019). Esse ambiente de modelagem unificado melhora a análise do sistema e reduz o número de conflitos que são comumente injetados em uma abordagem tradicional.

Nesse contexto, surge a engenharia de sistemas baseada em modelo (MBSE) que coloca os modelos como base do design (Cretu and Dumitriu, 2014). Através de uma abordagem top-down, fornecendo suporte aos requisitos, design, análise, verificação e validação associados ao desenvolvimento de sistemas complexos tais como as redes inteligentes.

Portanto, o objeto desta pesquisa é dar continuidade ao método proposto em (Orellana Postigo, 2021), que apresenta uma abordagem baseada na Engenharia de Sistemas Baseada em Modelos (MBSE) para modelar micro-redes, utilizando a norma IEC 61850 como arquitetura de referência. Nessa proposta, os requisitos seriam representados em um modelo orientado a objetivos (GORE), verificados formalmente usando Redes de Petri (Orellana Postigo, 2021). O processo de design preliminar é feito com um matching entre goals e sub-goals e algoritmos projetados em Redes de Petri. Este artigo mostra especificamente implementar um controlador em FPGA para automação de uma micro-rede, fechando o ciclo até o design preliminar (embodiement).

2 Descrição da proposta

A proposta deste trabalho está inserida em uma abordagem baseada no MBSE para modelar sistemas microgrid, utilizando a norma IEC 61850 como arquitetura de referência. O método proposto consiste em modelar os requisitos baseados em KAOS, compondo um modelo que pode ser formalizado em Redes de Petri, e, a partir deste modelo, propor o design preliminar deste sistema (Postigo et al., 2020), (Orellana et al., 2021). A modelagem dos requisitos e sua formalização foram descritos em trabalhos anteriores (Orellana Postigo, 2021), e esta proposta de continuidade fecha o ciclo até o design preliminar.

Na fase de design é feito um matching entre goals e sub-goals originados na fase de requisitos e algoritmos já projetados em Redes de Petri como requisitos gerais, e propostas de algoritmo (soluções) com transferência para hardware de lógica configurável (FPGA). A Figura 1 mostra o método proposto.

A ideia de utilizar FPGA, é devido as vantagens que este dispositivo oferece, tais como a flexibilidade, permitindo a programação e reprogra-

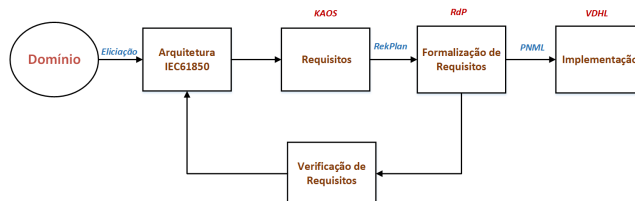


Figura 1: Método proposto
Fonte: elaborado pelo autor (2019)

mação por inúmeras vezes para atender a diferentes requisitos de design. Segundo (Tlelo-Cuautle et al., 2016), os FPGA consistem em uma matriz de milhares de blocos lógicos reconfiguráveis que podem ser ligados com outros blocos lógicos ou com blocos de I/O por meio das entidades de roteamento, conforme mostrado na Figura 2.

Além disso, os FPGAs são altamente paralelizáveis e podem ser configurados para executar várias tarefas simultaneamente, o que os torna ideais para aplicações que requerem alto desempenho e baixa latência, e também pode ser utilizado no controlador de uma microrrede.

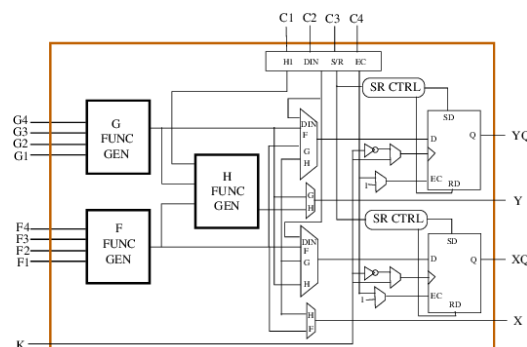


Figura 2: Arquitetura FPGA
Fonte: (Tlelo-Cuautle et al., 2016)

Portanto, o workflow gerado na fase de requisitos e representado em Redes de Petri gera um arquivo em *Petri Net Markup Language* (PNML) (Silva, 2016), que servirá de base para um arquivo no formato *Very High-Speed Integrated circuits Hardware Description Language* (VHDL), que é uma linguagem de descrição de hardware e permite a especificação da lógica dos circuitos em um alto nível de abstração, facilitando o desenvolvimento e a verificação do projeto. Nesse contexto, o VHDL reflete as funcionalidades dos objetivos delineados nos requisitos e as funcionalidades das soluções apresentadas no processo de matching.

Convém lembrar que, o PNML é uma descrição de uma rede de Petri em XML. Portanto, para realizar a conversão do PNML para VHDL é necessário transferir o modelo em uma descrição em VHDL que possa ser programada diretamente em um FPGA.

Assim, para transferir um modelo em PNML

para VHDL, usaremos a ferramenta PN2VHDL, que é uma ferramenta livre que converte Redes de Petri, em VHDL. Esta ferramenta foi escolhida por ser amplamente utilizada na área de sistemas embarcados para converter modelos de redes de Petri em código VHDL para implementação em FPGA. Entretanto, existem outras ferramentas disponíveis, como a CPN2VHDL e o PETRIFY.

O arquivo gerado será recompilado por uma ferramenta que gera algoritmos RTL (Register Transfer Level) que permitirá a gravação na FPGA. O código RTL é uma descrição de nível intermediário do comportamento do circuito digital em termos de operações de transferência de dados entre registradores e sinais lógicos, e é usado como entrada para a síntese lógica do circuito, descrevendo o seu comportamento em termos de sinais de entrada/saída e sinais internos. Como resultado final, se obtém o circuito integrado desejado conforme mostrado na Figura 3.

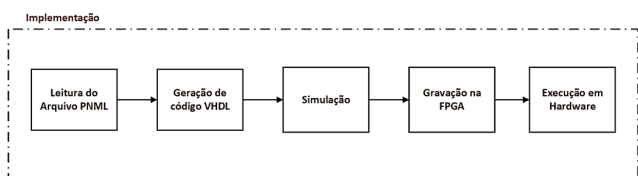


Figura 3: Implementação FPGA
Fonte: elaborado pelo autor (2022)

O design da plataforma de desenvolvimento segue as seguintes fases (que serão aplicadas ao estudo de caso):

- Conversão do arquivo PNML da Rede de Petri para um arquivo de descrição de hardware na linguagem VHDL,
- compilação do código VHDL,
- simulação,
- gravação,
- execução em hardware

Por outro lado, no desenvolvimento da ferramenta de conversão, realizado no ambiente integrado, a plataforma também possibilita o desenvolvimento da aplicação, pois engloba em um único ambiente o editor de código fonte, o compilador-ligador e o depurador. Além disso, possui interface gráfica com janela de gerenciamento dos arquivos do projeto.

Na geração do arquivo em formato VHDL, foi utilizada a plataforma de desenvolvimento de hardware configurável Quartus II, da Altera, para compilação do arquivo VHDL, simulação do arquivo RTL e gravação do código binário relativo ao hardware gerado na placa de desenvolvimento DE1, da Altera-Terasic, onde o modelo pode ser executado em hardware.

3 Aplicação do método proposto em um estudo de caso

Como estudo de caso aplicaremos o método proposto no controle de uma microrrede que fornece energia elétrica usando fontes de energia renovável. A revisão dos requisitos deste projeto, foi também objeto de trabalhos anteriores (Orellana et al., 2021), mais o matching de soluções é objeto somente deste trabalho. A ideia é que o mesmo método possa ser futuramente utilizado em microrredes de pequenas comunidades isoladas no interior da Amazônia.

Neste trabalho, objetiva-se utilizar o referido matching de soluções, comendadas anteriormente para implementar um controlador em FPGA a nível laboratorial, para automação de uma microrred. Portanto, seguindo o método proposto (Orellana et al., 2021), uma Rede de Petri e gerada, e esta representa os requisitos que derivada de um modelo KAOS mostrado na Figura 4, além disso, a automação da operação da microrrede basea-se na IEC 61850, conforme a Figura 5.

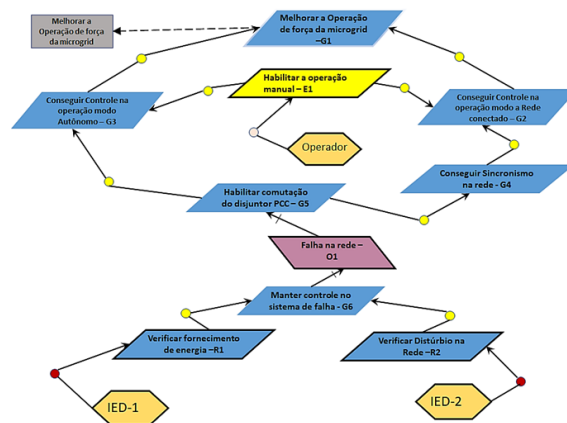


Figura 4: Modelo KAOS baseado na IEC-61850
Fonte: elaborado pelo autor (2019)

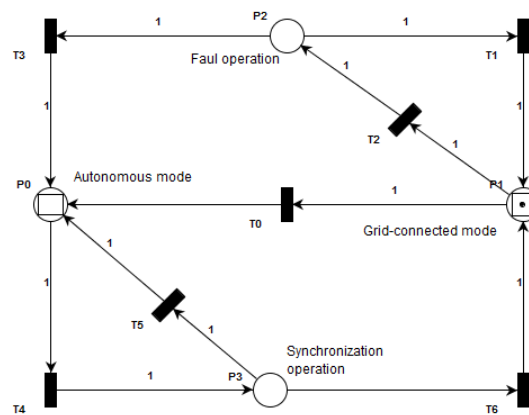


Figura 5: Rede de Petri baseado na IEC-61850
Fonte: elaborado pelo autor (2019)

Nesta figura apresenta-se a Rede de Petri uti-

lizando a ferramenta PIPE-3, mostrando a solução de controle para a microrrede, na qual, inicialmente foram identificados os seus principais componentes, como: geradores, inversores, baterias, carga e sistemas de comunicação.

Na etapa seguinte foram definidos os estados e transições do sistema. Estes estados incluem condições, como níveis de carga de bateria, demanda de energia, tensão e frequência. As transições representam os eventos que alteram o estado do sistema, como mudanças na geração de energia, desconexões ou reconexões dos diversos dispositivos.

O passo seguinte consiste na modelagem, ou seja, na modelagem das relações entre os elementos da microrrede, por exemplo, a intermitência das fontes de energia renováveis, como a carga de energia gerada pelo gerador é armazenada na bateria, como a demanda de energia afeta a carga da bateria, ou como a rede é afetada por eventos externos, como falhas na rede de energia.

A etapa final é a análise e verificação do modelo, utilizando a propriedades das Redes de Petri conforme Figura 6, Aqui, foram analisados os componentes conservativos e repetitivos da rede, ou seja, as invariantes da matriz de incidência e, equação de estados.

Assim, verifica-se que:

- todas as marcações podem ser alcançadas desde a marcação inicial, portanto, a rede é viva e está livre deadlock;
- A RdP representa adequadamente a microgrid (sem deadlok) implica que no haverá interrupção no fornecimento de energia;
- A rede é Limitada porque a árvore alcançabilidade também é limitada, indicando que o sistema é controlável;
- A análise de invariantes que mostra que o sistema é conservativo.

Na prática a análise do modelo da microrrede, permite determinar como as mudanças do sistema afetam o desempenho da microrrede e como se comporta sob diferentes condições de carga ou geração de energia. Isso ajudará a garantir que o modelo seja preciso e útil para garantir o desempenho da microrrede.

A transferência do modelo em rede de Petri para FPGA é feita usando a linguagem de descrição de hardware VHDL (VHSIC Hardware Description Language). Esta linguagem permite que o modelo em rede de Petri seja codificado em um formato que permite a programação direta no FPGA. Assim, o workflow gerado na fase de requisitos e representado em Redes de Petri gera um arquivo PNML. Ao salvar o arquivo na plataforma PIPE, é gerado um arquivo PNML, que é uma descrição da Rede de Petri em XML, mostrado na Figura 7.

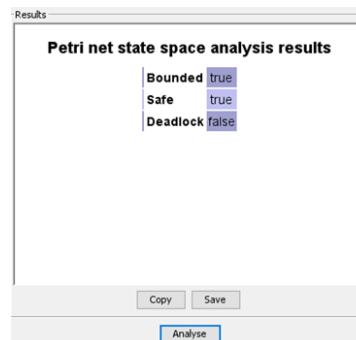


Figura 6: Resultado da Análise de propriedades da Rede de Petri

Fonte: elaborado pelo autor (2019)

Para ajudar na conversão do arquivo PNML para VHDL, usamos a ferramenta PN2VHDL, que converte redes de Petri em VHDL. Também possibilita a identificação dos estados e transições do modelo, e a criação de uma descrição em linguagem VHDL. Esta ferramenta foi escolhida por ser amplamente utilizada na área de sistemas embarcados para converter modelos de redes de Petri em código VHDL para implementação em FPGA.

```
<?xml version="1.0" encoding="ISO-8859-1"?><pnml>
<net id="Net-One" type="P/T net">
<token id="Default" enabled="true" red="0" green="0" blue="0"/>
<place id="P0">
<graphics>
<position x="765.0" y="240.0"/>
</graphics>
<name>
<value>P0</value>
<graphics>
<offset x="0.0" y="0.0"/>
</graphics>
</name>
<initialMarking>
<value>Default,1</value>
<graphics>
<offset x="0.0" y="0.0"/>
</graphics>
</initialMarking>
<capacity>
<value>0</value>
</capacity>
</place>
<place id="P1">
<graphics>
<position x="540.0" y="60.0"/>
</graphics>
<name>
<value>P1</value>
<graphics>
<offset x="0.0" y="0.0"/>
</graphics>
</name>
<initialMarking>
<value>Default,0</value>
<graphics>
<offset x="0.0" y="0.0"/>
</graphics>
</initialMarking>
<capacity>
<value>0</value>
</capacity>
</place>
<place id="P2">
<graphics>
<position x="540.0" y="435.0"/>
</graphics>
```

Figura 7: PNML-PetriNet

Fonte: elaborado pelo autor (2022)

No passo seguinte cria-se o projeto no software de desenvolvimento FPGA, que inclui a cri-

ação do arquivo VHDL, correspondente ao modelo em rede de Petri, e a definição das entradas e saídas da microgrid. O arquivo com o código VHDL completo é gerado no mesmo diretório onde está a ferramenta e apresenta extensão VHD. Este arquivo é utilizado para a implementação no FPGA. O código VHDL gerado é mostrado na Figura 8, e descrito a seguir em suas diversas partes.

```

library ieee;
use ieee.std_logic_1164.all; entity controller is port (
  clk : in bit; -- 50MHz proprio do kit DE2-115
  reset: in bit; -- Botão de reset para o acionamento do Autonomous mode a qualquer momento
  control : in bit_vector[1 downto 0]; -- Entradas U e UD que controlam os estados FaultOP e SyncOP
  dg : in bit; -- Geração distribuída
  bess : in bit; -- Sistema de armazenamento de energia
  utility : in bit; -- Rede elétrica

  i0 : out bit; -- Led indicador do GridMode
  i1 : out bit; -- Led indicador do FaultOP
  i2 : out bit; -- Led indicador do Autonomous
  i3 : out bit; -- Led indicador do SyncOP
);
end controller;
architecture main of controller is
  type macro is (microGrid, faultOP, autonomous, syncOP); signal state: macro;
begin
  operationMode : process(clk, reset)
  begin
    if (clk'event and clk = '1') then case state is
    WHEN microGrid => i0 <= '1';
    i1 <= '0';
    i2 <= '0';
    i3 <= '0';
    IF utility = '1' and dg = '1' and bess = '1' and reset = '1' and control = "00" THEN
    -- Está conectado com a rede elétrica, tem bateria e geração distribuída? GridMode state <= microGrid;
    ELSIF utility = '0' and reset = '1' and control = "00" THEN
    -- Algum problema na rede elétrica? FaultOP state <= faultOP;
    ELSE
    -- Botão de reset acionado = Autonomous IF reset = '0' then
    state <= autonomous; END IF;
    END IF;
    WHEN faultOP => i0 <= '0';
    i1 <= '1';
    i2 <= '0';
    i3 <= '0';
    IF control = "01" and utility = '1' and reset = '1' THEN
    -- Problema na rede elétrica resolvido? GridMode state <= microGrid;
    ELSIF control = "10" and utility = '0' THEN
    -- Problema na rede elétrica não resolvido? Autonomous state <= autonomous;
    ELSE
    -- Caso não aconteça nenhuma mudança nos estados
    state <= faultOp;
    END IF;
    WHEN autonomous => i0 <= '0';
    i1 <= '0';
    i2 <= '1';
    i3 <= '0';
    IF dg = '1' and bess = '1' and utility = '1' and control = "10" THEN
    -- Geração distribuída, bateria e rede ok? Modo de sincronização state <= syncOp;
    ELSE
    -- Caso o problema persista state <= autonomous;
    END IF;
    WHEN syncOp => i0 <= '0';
    i1 <= '0';
    i2 <= '0';
    i3 <= '1';
    IF control = "11" and bess = '1' and dg = '1' and utility = '1' THEN
    -- Tudo certo com a sincronização. GridMode state <= microGrid;
    ELSIF control = "00" and utility = '0' THEN
    -- Falha na sincronização com a rede state <= autonomous;
    ELSE
    -- Tempo de sincronização state <= syncOp;
    END IF;
    END CASE;
  end if;
  end process operationMode;
end main;

```

Figura 8: Código VHDL

Fonte: elaborado pelo autor (2022)

O próximo passo é a compilação do projeto para gerar um arquivo binário que a ser carregado na FPGA. Ou seja, o código VHDL gerado pela ferramenta PN2VHDL foi compilado e simulado no software Quartus II da Altera, e posteriormente gravado na FPGA utilizando-se o kit de desenvol-

vimento Cyclone II da Altera, mostrado na Figura 9.

Assim, o código em VHDL gerado foi compilado para geração de arquivos para simulação e posterior gravação na FPGA. Cabe destacar que, a compilação do projeto envolve a síntese do código VHDL, a implementação do design e a geração do arquivo binário.

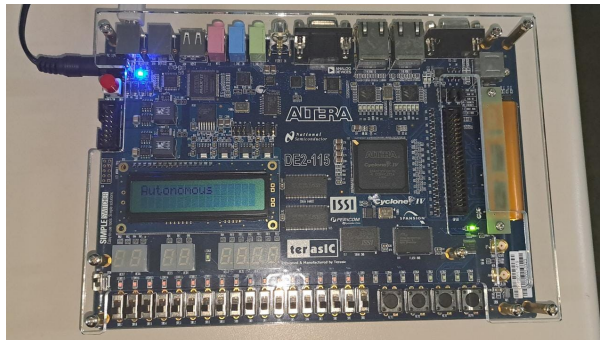


Figura 9: Kit FPGA Cyclone II Altera

Fonte: elaborado pelo autor (2022)

Para realizar a simulação utilizou-se a ferramenta disponível no software Quartus II. Verificase que com a ativação do sinal de Reset, os lugares da Rede de Petri recebem sua marcação nos lugares (P), e a evolução da marcação da Rede de Petri corresponde as bordas de subida do sinal de CLOCK.

O passo seguinte, corresponde à implementação do hardware, isto é, o arquivo binário gerado pelo software Quartus II a partir do código em VHDL é carregado na FPGA, com uso do software de programação de FPGA.

Na fase final, verifica-se o funcionamento do sistema microgrid na FPGA. Isto é, através do uso das ferramentas de depuração analisa-se o comportamento do sistema em diferentes condições para garantir que esteja funcionando corretamente.

Vale a pena destacar que ao transferir um modelo em rede de Petri para a FPGA devemos ter presente que a implementação do modelo em hardware eventualmente pode ser diferente do comportamento simulado do modelo em software. Por isso, o controle do sistema microgrid implementado na FPGA foi testado, verificando-se que este funciona de acordo com os requisitos gerados no KAOS e modelado na rede de Petri.

4 Conclusão

O equilíbrio entre produção e distribuição de energia elétrica aponta para a disseminação dos sistemas Smart Grid, caracterizados por serem abertos, distribuídos, heterogêneos e escaláveis. Esses sistemas exigem abordagens de design multidisciplinares e mais sofisticadas, que demandam uma

revisão dos métodos tradicionais (normalmente, puramente funcionais). Essa abordagem tradicional tende a ser substituída por métodos baseados em modelos sistêmicos, começando pela fase de requisitos. Como uma alternativa, os métodos prospectivos podem ser orientados a objetivos, ao invés de puramente funcionais, seguindo uma abordagem sistêmica - ao invés de focada em componentes - e focada no usuário final. Isso impõe uma disciplina orientada a serviços, mais do que para produtos. Além disso, essa nova abordagem tem a vantagem de se acoplar com os fornecedores e usuários, especialmente com um usuário final que é ao mesmo tempo provedor e consumidor (prosumer).

A proposta deste trabalho foi acoplar ao método proposto em (Orellana Postigo, 2021), que apresentou uma abordagem alternativa, baseada em modelos e em engenharia de sistemas, formalizada em Redes de Petri, para a fase de requisitos, tendo como referência a arquitetura IEC 61850. Entretanto, para obter agilidade que possa fazer frente à demanda já descrita, é preciso ter um *embodiment* bem delineado. Isso implica em ter um método que dê sequência ao processo de modelagem de requisitos e sua proposta formal para sintetizar protótipos rápidos que possam instanciar diferentes propostas de solução. Um processo de simulação pode ajudar a escolher uma das propostas, que deve seguir para a implementação final. O uso de FPGA's permite que esse processo seja plenamente acoplado à fase inicial de requisitos, permitindo ainda a automação do processo.

Além de sintetizar as Redes de Petri em código VHDL para a posterior gravação na FPGA's, o método pressupõe a existência de um isomorfismo com o workflow gerado na fase de requisitos, contribuindo para que se verifique a solução escolhida no **embodiment**. Este resultado da proposta foi comprovado mediante a implementação de um controlador para automação da microrrede.

Portanto, a aplicação do método proposto contribui para agilizar e melhorar o processo de modelagem de microrredes, bem como da sua prototipação, com foco em um novo tipo de usuário. Por outro lado, a possibilidade de ter vários sistemas orientados a serviço permitirá ainda sua reutilização e um futuro tratamento como sistemas de sistemas, trazendo benefícios ao ciclo de projeto.

Agradecimentos

Os autores agradecem suas respectivas instituições pelo apoio às suas atividades de pesquisa: Universidade do Estado do Amazonas (UEA), e a Escola Politécnica, da Universidade de São Paulo (USP).

Referências

- Cretu, L. G. and Dumitriu, F. (2014). *Model-Driven Engineering of Information Systems: Principles, Techniques, and Practice*, CRC Press.
- Falcão, D. M. (2010). Integração de tecnologias para viabilização da smart grid, *III Simpósio Brasileiro de Sistemas Elétricos* pp. 1–5.
- Orellana, M. A., Silva, J. R. and Pellini, E. L. (2021). A model-based and goal-oriented approach for the conceptual design of smart grid services, *Machines* **9**(12): 370.
- Orellana Postigo, M. A. (2021). *Método para o design integrado de sistemas smart grid: uma abordagem sistêmica baseada em modelos*, PhD thesis, Universidade de São Paulo.
- Postigo, M. A. O., Silva, J. M. and Silva, J. R. (2020). Applying goal-oriented requirement engineering and reference architecture for microgrid systems, *Brazilian Applied Science Review* **4**(3): 913–926.
- Postigo, M. A. O. and Silva, J. R. (2018). Modeling in petri nets for micro smart grid operation based on iec 61850 architecture, *2018 Simposio Brasileiro de Sistemas Eletricos (SBSE)*, IEEE, pp. 1–6.
- Silva, J. M. (2016). Using gore method for requirement engineering of planning & scheduling, *The 26th International Conference on Automated Planning and Scheduling*, p. 137.
- Tlelo-Cuautle, E., De La Fraga, L. and Rangel-Magdaleno, J. (2016). Engineering applications of fpgas, *Engineering Applications of FPGAs*.
- Uslar, M., Rohjans, S. et al. (2019). Applying the smart grid architecture model for designing and validating system-of-systems in the power and energy domain: A european perspective, *Energies* **12**(2): 258.