

METODOLOGIA PARA DESENVOLVIMENTO DE SISTEMAS EVOLUTIVOS - MEDSE: PRINCÍPIOS BÁSICOS

HIRAM C. C. AMARAL, RAFAEL S. MENDONÇA, ANDRÉ L.D. CAVALVANTE, VICENTE F. LUCENA JR.

*Programa de Pós-graduação em Engenharia Elétrica, Universidade Federal do Amazonas
Av. General Rodrigo Octávio Jordão Ramos, 6200, Setor Norte, Coroadó I -- Manaus, AM, Brasil
{hiramamaral, mendoncarms, andrecavalcante, vicente}@ufam.edu.br*

NASSER JAZDI-MOTLAGH

*Institute of Industrial Automation and Software Engineering – University of Stuttgart
Pfaffenwaldring 47, 70550 – Stuttgart – Germany
nasser.jazdi@ias.uni-stuttgart.de*

Abstract— Emerging manufacturing paradigms have been largely used in an attempt to solve the customization problem of mass production - also known as the problem of low batches with high levels of variety of products. Cyber-Physical Systems (CPS) and Internet of Things (IoT) architectures not only reinforce research work towards the customization solution, but also create conditions for industrialized nations to meet the requirements of the Industry Platform 4.0. This paper presents the bases of a Methodology for Development of Evolvable Systems (MeDSE) that can be applied in both engineering systems (automation systems) and in computer systems (cyber physical systems). The methodology is composed of ten stages of development. Each step starts from a global problem, which is divided into regional problems that are addressed until they are local problems. The methodology applied in the development of a prototype called System Intelligent Evolvable Process Agile (SIAPE). The results demonstrated that the methodology adheres to the recommendations of the Industry 4.0 Platform.

Keywords— Automation manufacturing, Evolvable Production System, Industry 4.0, CPS e MeDSE.

Resumo— Paradigmas emergentes de fabricação têm sido largamente usados na tentativa de solucionar o problema da customização de massas, também conhecido como o problema dos lotes baixos com elevados níveis de variedades de produtos. Notadamente os Evolvable Production Systems (EPS) têm conseguido tratar o problema. Os Cyber-Physical Systems (CPS) e as arquiteturas Internet of Things (IoT) reforçam não somente as pesquisas rumo à solução da customização, mas criam condições para as nações industrializadas se adequarem às exigências da Plataforma da Indústria 4.0. Este trabalho apresenta as bases de uma Metodologia para Desenvolvimento de Sistemas Evolutivos (MeDSE) que pode ser aplicada tanto em sistemas de engenharia (sistemas de automação) quanto em sistemas de computação (cibersistemas). A metodologia é composta por dez etapas de desenvolvimento. Cada etapa parte de um problema global, que é dividido em problemas regionais que são tratados até que sejam transformados em problemas locais. A metodologia foi aplicada no desenvolvimento de um protótipo denominado de Sistema Inteligente Ágil de Processo Evolutivo (SIAPE). Os resultados demonstraram que a metodologia está aderente às recomendações da Plataforma da Indústria 4.0.

Palavras-chave— Automação Industrial, Sistemas Evolutivos de Produção, Indústria 4.0.

1 Introdução

A busca de soluções para o problema da customização, isto é, a produção de produtos em lotes pequenos e com alta variabilidade de produtos, está unindo tecnologias de diversas áreas como a Internet e a Automação, que aproximam mercados e consumidores de locais distantes dentro do mercado global e transformam a economia mundial numa grande rede de conhecimento (Abele, 2008).

Essa busca tem aproximado, particularmente, as áreas da Engenharia e da Computação em pesquisas envolvendo a manufatura industrial (Rocha, 2014).

Como resultado dessas pesquisas, vários novos paradigmas de produção têm realizado várias tentativas de solução. Pode-se citar o paradigma *Flexible Manufacturing System* (FMS) (Hoda, 2006) que tratou o problema de personalização, isto é, produção de produtos em lotes médios e com média variedade de

produtos. Uma vez o consumidor tendo acesso à algum tipo de personalização, buscou ainda mais especificidades nos produtos, o que levou a um aumento na variedade de produtos e a uma diminuição do número de produtos por lote, isto é, levou à customização em massa.

Outros paradigmas, tais como *Bionic Manufacturing System* (BMS) (Ueda, 1992), *Holonic Manufacturing System* (HMS) (Christensen, 1994), *Reconfigurable Manufacturing System* (RMS) (Brussel, 1999) e *Evolvable Production System* (EPS) (Onori, 2002), apareceram nas últimas décadas e apresentam possíveis soluções para os problemas da personalização e da customização.

Neste trabalho, portanto, há uma diferença conceitual entre personalização e customização. O primeiro se refere à produção de produtos em lotes médios e com pouca ou média variabilidade. O segundo conceito se refere à produção de produtos em pequenos lotes e com alta variabilidade de produtos.

As soluções propostas por esses novos enfoques da manufatura, na sua maioria, são desenvolvidas com diversas ferramentas de inteligência artificial, linguagens de programação amigáveis e metodologias que facilitam a criação de sistemas que são utilizados em áreas distintas. Todavia ainda há uma lacuna em uma metodologia de desenvolvimento para EPS.

EPS trata o chão-de-fábrica e a empresa como um todo através do paradigma de multiagentes, trazendo técnicas de IA e as cyber estruturas para dentro da manufatura.

Este trabalho propõe uma Metodologia para Desenvolvimento de Sistemas Evolutivos (MeDSE) em que os princípios do paradigma EPS participam desde as fases mais básicas até o final do desenvolvimento. Como experimento realizou-se o Sistema Inteligente Ágil de Processo Evolutivo - SIAPE, um sistema mecatrônico baseado em agentes inteligentes que, no paradigma, recebe o nome de agentes mecatrônicos.

Este trabalho está assim organizado: a seção II descreve principais pontos que identificam o estado da arte onde este trabalho está inserido, começando por identificar a criação dos paradigmas que antecederam o paradigma EPS; a seção III descreve a Metodologia para Desenvolvimento de Sistemas Evolutivos (MeDSE); a seção IV apresenta os procedimentos da MeDSE que realizaram o Sistema Inteligente Ágil de Processo Evolutivos - SIAPE e; na seção V são analisados os resultados do experimento SIAPE. Na seção VI são colocadas as nossas conclusões.

O MeDSE tem como objetivo a criação de sistemas evolutivos (EPS), contudo, como aqui mostrado, serve igualmente para o desenvolvimento de atualizações de sistema legados para a nova realidade da Indústria 4.0.

2 Estado da Arte

O lançamento das arquiteturas da *Internet of Things* (IoT) com seus protocolos de alta performance e precisão (Rotondi, 2013) vislumbram contribuições significativas que estão sendo combinadas com as pesquisas em torno dos paradigmas EPS. Neste caso, a abordagem é chamada de *Cyber-Physical Systems* (CPS), que visa a realização das cyberinfraestruturas descritas em Krogh, (2008).

Tais visões da automação estão no centro das transformações dos sistemas nas diversas áreas do conhecimento, que envolvem propriedades como auto-organização e autoaprendizagem, as chamadas auto-x, interoperabilidade e interconectividade, as chamadas inter-y, e alta flexibilidade e alta-escalabilidade (alta-z), que levarão à concretização da Plataforma da Indústria 4.0 (Drath, 2014).

As transformações que tais visões trazem à manufatura produziram um ambiente que neste artigo é chamado de Indústria Híbrida, por aliar conceitos e paradigmas antigos com os novos. Tais paradigmas são comentados nas subseções a seguir.

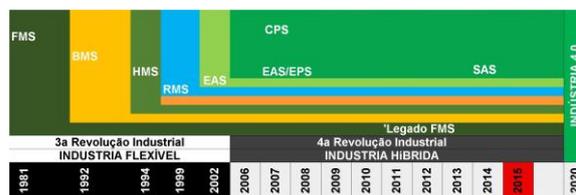


Figura 1. Paradigmas de Produção

2.1 Paradigmas da manufatura

A Figura 1 registra a evolução dos paradigmas de manufatura no tempo, indo do que chamamos Indústria Flexível para a Indústria Híbrida e alcançando o que se chama de Indústria 4.0.

Tudo começa no início dos anos 80, com o advento das máquinas flexíveis, que conseguiu-se tratar o problema da produção em lotes médios e pequenos com o paradigma FMS.

A partir de 1992, aparece o paradigma BMS, baseado no comportamento dos sistemas biológicos, e sua forma hierarquizada de tratar as questões orgânicas. Quando aplicados ao sistema produtivo, consegue-se gerar sistemas de manufatura com as características comuns aos corpos biológicos, tais com adaptabilidade e auto-organização.

Ainda na década de 90, lançado por James Christensen, aparece o paradigma HMS (Christensen, 1994), baseado no conceito de *hólons*. *Holon* é uma palavra grega que significa o Todo ou a Parte.

Ao final daquela década, Yoram Koren lança uma abordagem, chamada de RMS (Yoram, 1999) como forma de manter a competitividade das empresas por meio de sistemas de manufatura que tentam ser sensíveis a todas as mudanças de demanda do mercado.

Já no início dos anos 2000, Onori propõe o paradigma EAS. Baseado em Sistemas Multiagentes (SMA) (Wooldridge, 2002) serve de base para os chamados sistemas evolutivos, isto é, sistemas completamente responsivos em que a entrada ou retirada de elementos do sistema fazem parte do seu próprio funcionamento.

Em 2006 Onori e Barata (Onori, 2006) apresentam as bases do paradigma EPS. EPS é uma extensão de EAS, portanto utiliza o conceito de agentes inteligentes para realizar as operações dentro do sistema produtivo.

Mais recentemente, surge o *Symbiotic Assembly System* (SAS) integrando os sistemas convencionais de automação às capacidades do ser humano (Ferreira, 2014).

2.2 Metodologias e desenvolvimento

Concomitantemente com o desenvolvimento de novos paradigmas e visões da automação e manufatura, as metodologias de desenvolvimento de sistemas de automação também evoluíram ao longo do tempo.

Em 1997, com o lançamento da UML (*Unified Modeling Language*) foi possível o uso de uma fer-

ramenta para apoiar a prática de engenharia de software orientada a objetos. A UML também serve de linguagem de modelagem padrão para sistemas concorrentes e distribuídos, o que extremamente necessário para o ambiente industrial (Corporation, 1997).

Em 1988, Meyer criou um modelo orientado à objetos a partir de uma analogia com o sistema econômico: os agentes de software são chamados de fornecedores e clientes, e os protocolos de comunicação entre os agentes são chamados de contratos. (Meyer, 2000).

Nos anos 2000, Deloach (2001) publicou uma metodologia mais estruturada denominada de MaSE, que adicionou à abordagem estruturada tradicional, as ferramentas baseadas em agentes. Já Caire (2001) divulga a Metodologia MESSAGE que utilizou mensagens como o cerne principal do sistema. Em suas ideias, o desenvolvimento dos sistemas apresenta diferentes níveis de abstração, sendo o nível 0, relativo aos *stakeholders* e ao meio ambiente, e o nível 1 relativo às relações entre agentes inteligentes.

Mais recentemente, Abdelaziz (2010) lançou a Metodologia *Multi-Agent Development System* (MASD) que fornece aos designers de sistemas baseados em agentes um conjunto de métodos e orientações para a construção de sistemas complexos, baseados em sistemas multiagentes, de uma forma detalhada e completa.

2.3 Plataforma da Indústria 4.0

O *white paper* da Indústria 4.0 (i4.0) (Bitkom, 2015) formulou as questões centrais da i4.0, a partir de um ponto de vista da pesquisa e inovação, conforme segue:

Integração vertical (IVe) é caracterizada pela capacidade de os sistemas integrarem os seus processos técnicos aos processos de negócio da empresa, como forma de estratégia empresarial.

Integração horizontal (IHo) é a agregação, em tempo real, dos elementos de conhecimento que esses sistemas geram no chão-de-fábrica.

Integração humana (IHu) é caracterizada pelo uso das capacidades de criação dos seres humanos na recuperação dos sistemas legados (aqui chamados de i3.0).

Engenharia 4.0 (E4.0) é denotada pelos domínios dos conhecimentos de Engenharia durante todo o processo do ciclo de vida do produto, desde a concepção ao encerramento do projeto.

2.4 Sistemas Evolutivos de Produção

Os sistemas evolutivos são baseados em agentes inteligentes e autônomos, detentores de habilidades (*skills*) e que são capazes de cooperarem entre si para a realização dos objetivos propostos ao sistema. Esta cooperação leva tais sistemas a possuírem a capacidade de adaptação e de evolução (Rosa, 2013).

Por adaptação entende-se que o sistema é capaz de propor uma configuração alternativa de si mesmo para minimizar os efeitos adversos de perturbações. Adaptação é de curto prazo e, normalmente, implica auto reconfiguração na forma de ajustes de parâmetros. Já o termo evolução refere-se ao sistema que é capaz de permitir a introdução ou remoção de módulos existentes sem implicar na quebra do seu funcionamento. A evolução se caracteriza num processo de longo prazo, podendo o sistema evoluir até o limite da tecnologia ou ao da planta fabril.

Por causa da sua arquitetura baseada em agentes inteligentes, que possuem a capacidade de auto-organização, possuem também a propriedade do “plugar e produzir” (Barbosa, 2010). Tais sistemas estão sendo desenvolvidos e testados objetivando as fábricas inteligentes e a padronização proposta pela Plataforma da Indústria 4.0.

3 Metodologia para Desenvolvimento de Sistemas Evolutivos

A Metodologia para Desenvolvimento de Sistemas Evolutivos - MeDSE nasceu da necessidade da criação de um procedimento que contemplasse a noção de agentes inteligentes e seus *skills* no paradigma EPS, desde as primeiras fases do desenvolvimento. Depois de criado, o método foi comparado aos métodos utilizados por Rumbaugh (Rumbaugh, 1997) e Larman (Larman, 2007) para mostrar a aderência da MeDSE com metodologias já estabelecidas.

A MeDSE é formada por três fases de desenvolvimento, a saber: Fase de Conceitos, Fase de Realizações e Fase de Finalizações. A seu turno, cada fase é composta por um número parcial de etapas, de um total de dez, que facilitam o desenvolvimento de cada fase. Essas fases são descritas nas próximas seções.

3.1 Conceitos importantes para MeDSE

Para verificar a viabilidade de desenvolvimento da MeDSE os seus procedimentos foram utilizados, primeiramente, na elaboração de um projeto que reproduziu um sistema flexível básico. A partir desse projeto de sistema flexível, foram estudados alguns conceitos que foram aplicados no desenvolvimento de um sistema evolutivo básico. O primeiro projeto foi chamado de Produto UFAM, o segundo foi denominado de Sistema Inteligente Ágil de Processo Evolutivo - SIAPE. Esta subseção descreve esses projetos e os conceitos básicos para a aplicação da MeDSE.

Produto UFAM é um sistema de carimbos de letras desenvolvido para a montagem de anagramas a partir de um CLP e módulos pneumáticos, tal como os sistemas de montagem atuais. Esse sistema foi desenvolvido especificamente para representar os sistemas da i3.0. Este primeiro protótipo originou a parte de hardware do SIAPE com as devidas melhorias e restrições impostas pelo projeto.

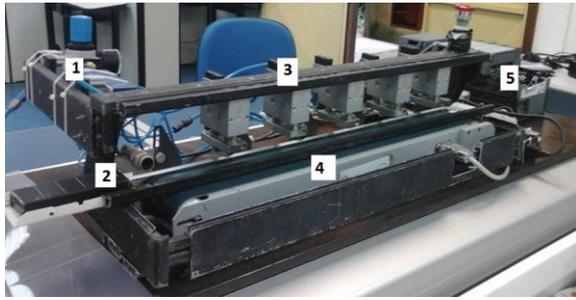


Figura 2. Produto UFAM

Portanto, algumas das características do SIAPE são herdadas tanto de ambientes i3.0, quanto de ambientes i4.0. Na Figura 2 estão identificados os seus componentes básicos: 1. Controle pneumático; 2. Sensores; 3. Eletroválvulas; 4. Esteira e 5. CLP.

Sistema Inteligente Ágil de Processo Evolutivo (SIAPE) é uma planta composta de quatro subsistemas que funcionam integradamente para reproduzir as duas características básicas dos sistemas EPS: a capacidade de adaptação, implicando que o sistema deve ser capaz de reagir por meio de autoconfiguração para reduzir os efeitos das perturbações ocasionadas no ambiente do chão de fábrica, e a capacidade de evolução, implicando que o sistema deve ser capaz de permitir a introdução ou alteração de novos módulos existentes em tempo de processamento sem gerar paradas consideráveis nas linhas de produção. A MeDSE foi utilizada para realizar o desenvolvimento do SIAPE.

Outros conceitos importantes:

Agentes mecatrônicos são os agentes inteligentes que executam nos módulos do sistema de manufatura. Tais módulos são identificados, no MeDSE pela sigla MESC, de Mecânica, Eletrônica, Software, Comunicação, isto é, as partes que formam um sistema/subsistema/módulo completamente funcional.

Aplicado à noção de agentes inteligentes, que são executados em módulo mecatrônicos, o MeDSE propõe a estrutura mostrada na Figura 3, que ilustra os conceitos de Problema Global (PG), Problema Regional (PR) e Problema Local (PL) dos módulos MESC.

Problema Global (PG) é definido como a identificação do sistema como um todo e todas as partes a serem desenvolvidas. Por exemplo, um carro com suas partes mecânicas (M) acopladas a um motor elétrico (E) é integrado a um controlador, o qual contém um software (S) que tem a capacidade de acelerar e parar o carro conforme ordens vindas pela rede (C).

Problema Regional (PR) é definido como uma visão do sistema na qual os relacionamentos entre as partes que o formam dependem funcionalmente uma das outras. Aqui são tomados os atributos/propriedades importantes das partes que foram o PR. Na Figura 3, por exemplo, ME e MES correspondem a dois problemas regionais. ME contém atributos/propriedades tanto mecânicos quanto eletrônicas, que formam então uma entidade de nível mais alto, integrada.

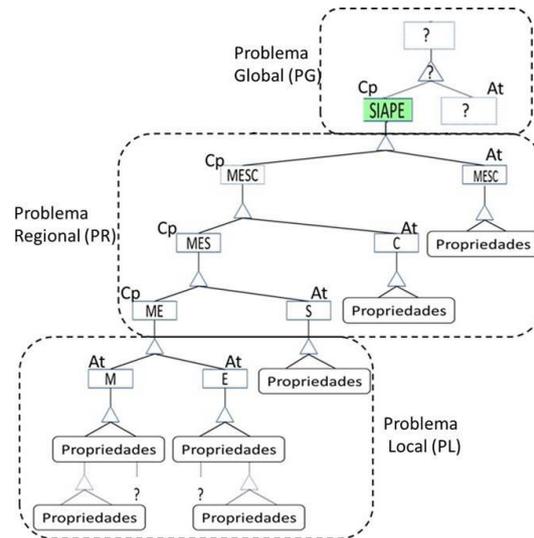


Figura 3. MeDSE: PG, PR e PL

MES, por outro lado, contém atributos/propriedades mecânicas, eletrônicas e de software. Formam uma entidade de nível ainda mais alto, quase um módulo completo. Já MESC é outro problema regional e que denota um módulo completo, um agente mecatrônico.

Problema Local (PL) é definido como a visão em que as entidades que formam o sistema são atômicas, isto é, podem ser analisadas e desenvolvidas sem interferência de outras entidades. Na Figura 3, as entidades M e E, que correspondem aos conjuntos Mecânicos e Eletrônicos, podem ser analisadas individualmente com seus atributos e propriedades eletrônicas e mecânicas separadamente.

Ainda analisando a Figura 3, pode-se perceber que as entidades M e E não conseguem mais ser decompostas em outras entidades, pois encontram-se em suas formas atômicas. Quando uma entidade atinge a sua forma atômica, o projetista chegou ao nível ideal para realizar a modelagem da entidade, pois em sua forma atômica, a entidade não sofre interferências de outra entidade externa.

Assim, no desenvolvimento de um sistema de automação, pode-se partir dessas entidades atômicas (PL), depois, integradas e testadas de acordo com o plano estabelecido (PR) e, por fim, temos o sistema completo (PG).

Modularidade é definida como a capacidade que o sistema tem de ser construído em módulos. Por exemplo, para produzir uma palavra, com quatro letras (“UFAM”), num pedaço de papel pode-se desenvolver alguns sistemas com vários níveis de modularidade:

Exemplo 1. Pode-se produzir um sistema com um carimbo com as letras “A”, “F”, “M” e “U” já dispostas na posição correta e formar a palavra “UFAM”;

Exemplo 2. Pode-se produzir um sistema com quatro carimbos com a letra “A”, “F”, “M” e “U” separadamente e o sistema decide carimbar a palavra UFAM ou outra palavra com as letras disponíveis;

Exemplo 3. Pode-se produzir quatro carimbos: um com traço na vertical ("|"), outro com um traço na horizontal ("-") e os outros dois com traços inclinados ("/" e "\"), e, então, carimbar os traços necessários para se produzir a palavra "UFAM";

Note-se que, para cada exemplo acima, há repercussões na decisão. O exemplo 1 é o mais simples de ser realizado, entretanto, é o menos flexível. Uma mudança ambiental simples, como por exemplo, a mudança no anagrama a ser impresso (o produto sofreu uma personalização), exige um retrabalho no sistema produtivo, isto é, um outro carimbo deve ser providenciado.

Lembrar que o exemplo do Produto UFAM e do SIAPE são acadêmicos, mas refletem os atributos de sistemas de automação industriais reais.

No paradigma EPS, o primeiro exemplo pode ser implementado com dois agentes: o primeiro, chamado de cognitivo, apenas informará quando o outro agente, chamado de agente motor, deve carimbar a palavra "UFAM".

Por outro lado, o segundo exemplo é mais complexo, porém mais flexível. De fato, qualquer dos anagramas possível a partir das letras dos possíveis ("A", "F", "M" e "U") pode ser montado.

No paradigma EPS, tal sistema pode ser implementado também com 2 tipos de agentes diferentes, um cognitivo e os outros motores. O agente cognitivo contém as informações necessárias para gerar os anagramas. Os agentes motores estão associados a cada um dos carimbos e, portanto, a cada uma das letras disponíveis. Aqui, além da informação de que letra deve ser carimbada, o agente cognitivo deve informar ao agente motor a posição da letra para formar a palavra que se quer ("UFAM").

No último exemplo, pode-se virtualmente carimbar todas as letras e números disponíveis. É a solução mais flexível, contudo a solução mais complexa e cuja a execução, mesmo de pequenos anagramas, exija que a máquina opere várias vezes a fim de desenhar as letras, de acordo com padrões estabelecidos.

No caso do paradigma EPS, também pode-se implementar essa solução através de agentes cognitivos que possuem a informação da construção das letras e anagramas e de agentes motores correspondentes aos traços. Um agente cognitivo pode continuar a informar a letra e a posição, contudo, um outro agente cognitivo deve ser capaz de imprimir qualquer letra a partir de chamadas aos agentes motores. Estes, por sua vez, devem ser capazes de fazer o carimbo de seu traço.

Analisando o conceito de modularidade, o sistema do exemplo 2, em relação ao sistema do exemplo 1, tem um nível de modularidade maior, pois existe a possibilidade de escolha da letra (módulo) a ser carimbada. Já o sistema do exemplo 3, em relação ao sistema do exemplo 2, é também mais modular.

Contudo, uma outra forma de implementar o sistema do exemplo 3, seria através de dois tipos agentes, um cognitivo, responsável pelo anagrama e os

outros motores, responsável pelos traços. Neste caso, o agente cognitivo deve saber como cada letra é formada para emitir os comandos corretos para cada agente motor.

Isto nos leva a um outro conceito: o nível de granularidade, isto é, a qualidade e quantidade de informações para se produzir um módulo.

Granularidade (Onori, 2011) é definida como a qualidade da informação que um agente deve possuir sobre os demais agentes e a quantidade dessas informações, a fim de se conseguir um objetivo definido.

A granularidade fina implica na implementação de um grupo de agentes que fica responsável por partes de um objetivo. Por exemplo, a impressão de uma letra para fazer um anagrama completo, no caso do exemplo 3 com um agente cognitivo, implica que mais conhecimento (quantidade), e de mais baixo nível (qualidade), é necessário ao agente para realizar a sua função. Já no caso do exemplo 3 com dois agentes cognitivos, um deles encapsula parte da complexidade, deixando a granularidade mais grossa. O caso do exemplo 2, acontece a mesma situação, onde o agente cognitivo (único) contém informações de nível mais elevado. A granularidade grossa implica que um módulo possui um nível de informação tal que é possível se definir anagramas completos a serem realizados e não partes desses.

Sistemas EPS apresentam também uma característica herdada dos multiagentes, a emergência. Sistemas que possuem emergência são ditos sistemas emergentes.

A emergência aqui é entendida como a propriedade de um sistema ser construído de forma determinística, mas que, por meio do conceito da auto-organização (Frei, 2007), apresenta a propriedade de adquirir uma estrutura espacial, temporal ou funcional sem interferência externa (Barbosa, 2010). Em outras palavras, os módulos do sistema são concebidos de forma determinística, isto é, cada um é desenvolvido com a capacidade de realizar um ou poucos objetivos, entretanto, o funcionamento do sistema como um todo, a sequência temporal ou espacial de execução, não necessariamente vai ser previsível.

3.2 Fase de Concepção

A Fase de Conceitos corresponde à realização conceitual, isto é, a concepção do sistema e a definição dos requisitos gerados a partir das necessidades do cliente, por requisitos mandatórios ou relacionados aos sistemas evolutivos. Seus procedimentos partem da consideração de que existe um problema global que deve ser subdividido em problemas menores para serem tratados atômica e até se tornarem requisitos que são modelados e simulados para que possam garantir a definição das especificações das partes do sistema. Essa fase é composta por duas etapas ilustradas na Figura 4 e descritas a seguir.

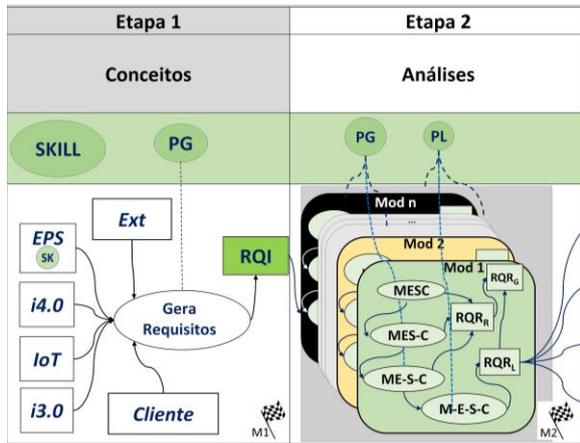


Figura 4. Fase de Concepção

- Etapa 1 Conceitos: tem como entrada as informações relativas ao sistema e as necessidades do cliente e sua saída é um documento de definição da concepção do sistema que contém os Requisitos Iniciais (RQI) que devem ser aprovados por ambas as partes, desenvolvedores e cliente.

- Etapa 2 Análises: aqui os RQI são analisados como Problemas Globais (PG) e refinados em problemas regionais (PR) até que se consiga um nível atômico denominado de problema local (PL). Além de se atingir o nível atômico relacionados às aplicações Mecânicas (M), Elétricas (E), Software (S) e Comunicação (C) esses problemas são analisados considerando as habilidades dos agentes mecatrônicos do paradigma EPS (*skills*) e são evoluídas nas etapas seguintes.

3.3 Fase de Realizações

Na Figura 5, corresponde aos processos práticos do desenvolvimento, isto é, a realização das etapas de Projeto de Sistema, Projeto de Classes, Implementações e Testes, conforme Rumbaugh (RUMBAUGH, 1991) e à Etapa de Projeto de Larman (LARMAN, 2007). Nesta fase os requisitos refinados são modelados, simulados, transformados em projetos a partir das especificações técnicas, integrados e validados modularmente.

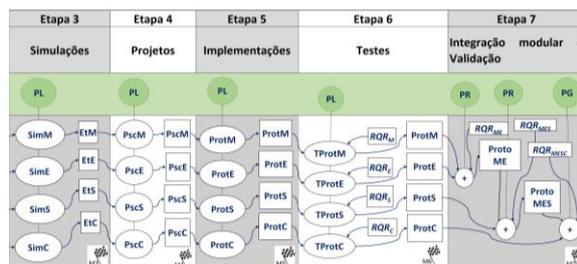


Figura 5. Fase de Realizações

- Etapa 03 Simulações: os requisitos refinados do sistema em forma de problema global (RQRs), na forma regional (RQRr) e local (RQRl) são modelados e simulados (SimM, SimE, SimS e SimC) até que estes possam ser garantidos para serem confirmados como Especificações Técnicas (Et) de cada problema

local, a saber: Especificação Técnica Mecânica (EtM), Especificação Técnica Elétrica (EtE), Especificação Técnica de Software (EtS) e Especificação Técnica de Comunicação (EtC).

- Etapa 04 Projetos: As especificações técnicas são realinhadas na forma de projetos parciais para formarem o Projeto de Sistema e de Classes de cada problema local (PscM, PscE, PscS e PscC).

- Etapa 05 Implementações: Os projetos de cada problema local são implementados, marcando as realizações iniciais dos protótipos locais: Os protótipos da parte mecânica (ProtM), da parte eletrônica (ProtE), da parte de software (ProtS) e da parte de comunicação (ProtC) tornam-se plantas reais dos projetos locais.

- Etapa 06 Testes: Os protótipos construídos são submetidos aos processos de teste (TProtM, TProtE, TProtS e TProtC) contra os requisitos refinados de cada problema local (RQRm, RQRr, RQRs e RQRc). Após a aprovação são levados à condição de protótipos testados e aprovados para o processo de integração.

- Etapa 07 Integração e validação modular: Os protótipos testados são submetidos ao processo de integração e validação modular: O protótipo mecânico (ProtM) é integrado ao protótipo eletrônico (ProtE) o qual denomina-se protótipo ME (ProtoME), este é submetido ao processo de validação modular contra os requisitos refinados da entidade ME (RQRme). O ProtoME é integrado ao protótipo da parte de software (ProtS) e a essa integração denomina-se por MES (ProtoMES). O protótipo MES é então submetido ao processo de validação modular contra os requisitos refinados da entidade MES (RQRmes). O ProtoMES é então integrado ao protótipo da parte de comunicação (ProtC) e com essa integração chega-se à realização da solução do problema global MESC.

3.4 Fase de Finalizações

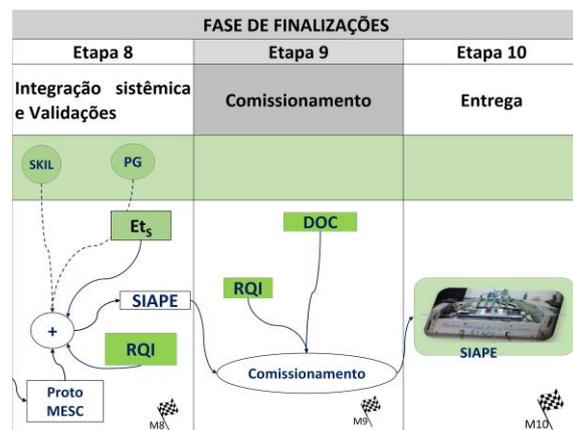


Figura 6. Fase de Finalizações

A Fase de Finalizações mostrada na Figura 6 tem como principal objetivo verificar e validar a integração sistêmica do sistema desenvolvido para posterior comissionamento e entrega. A verificação neste traba-

lho é entendida como a atividade que analisa se os requisitos funcionais e não-funcionais foram atendidos, enquanto a validação analisa se as necessidades do cliente foram atendidas.

Após a realização dessas atividades, o sistema estará preparado a realizar um plano de produção que será solicitado por um usuário do sistema utilizando o manual técnico. Ao se cumprir a meta (a realização do plano) os agentes mecatrônicos do sistema estarão realizando suas habilidades (*skills*) evidenciando que os requisitos do cliente foram atendidos.

- Etapa 08 Integração e validação sistêmica: Nesta etapa o sistema é solicitado a realizar um pedido de usuário que represente uma situação normal de produção. A realização do pedido evidenciará, neste projeto, as atividades dos agentes inteligentes mecatrônicos que se utilizam de seus *skills* para atingir as metas impostas pelo usuário (a realização do pedido).

- Etapa 09 Commissionamento: A documentação técnica e do usuário é finalizada e o sistema é revisado comparando com o manual de operação técnica e com o manual de usuário.

- Etapa 10 Entrega: O sistema é demonstrado ao cliente e entregue oficialmente, encerrando a fase de desenvolvimento.

4 Aplicação da MeDSE no desenvolvimento do SIAPE

Esta seção descreve uma implementação da MeDSE por meio do desenvolvimento de um sistema denominado de Sistema Inteligente Ágil de Processo Evolutivo (SIAPE) que é formado por uma planta que visa a reproduzir as características básicas dos sistemas EPS: Adaptabilidade e evolutividade.

Fase de Conceitos: Seguindo a metodologia MeDSE, o Problema Global (PG) foi definido com base nos sistemas EPS, na Indústria 4.0, no Produto UFAM que representou um sistema legado, nos requisitos de segurança, economia e globalização e nas restrições estabelecidas. Ao final da Fase de Conceitos definiu-se o documento de projeto contendo os requisitos e as especificações de hardware e software que concretizariam as partes lógicas e físicas do sistema.

Fase de Realizações: A partir deste ponto todas as etapas da Fase de Realizações foram processadas e realizadas. Da modelagem seguiu-se as simulações que geraram os circuitos e modelos como especificações que foram organizadas e receberam o status de projeto de sistema.

O projeto de sistema foi transformado em simulações (tanto na parte de hardware quanto na parte de software) que foram transformadas em protótipos, e esses foram testados e validados, modular e sistemicamente, por meio da verificação das especificações técnicas e requisitos e todos os dispositivos foram ligados à rede para que o protótipo MESC fosse validado modularmente. A Figura 7 ilustra o resultado da

final da fase de realizações do Sistema Inteligente Ágil de Processo Evolutivo – SIAPE.

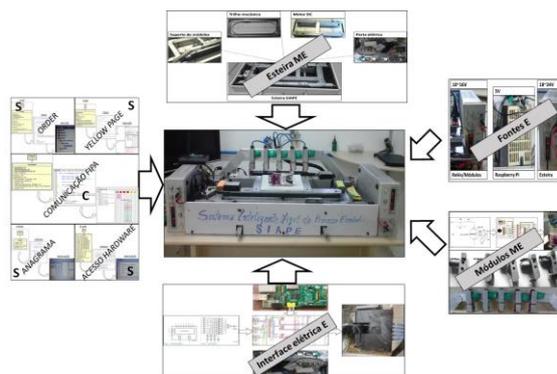


Figura 7. MeDSE Fase de Finalizações

Fase de Finalizações: O protótipo MESC validado modularmente foi submetido ao processo de validação sistêmica, aqui entendida como a validação que realiza um estudo de caso preparado pela equipe de desenvolvimento com o objetivo de evidenciar as funcionalidades e performance do sistema e deverá ser reproduzida em parte ou no todo.

5 Resultados

O SIAPE, desenvolvido pelo Método de Desenvolvimento de Sistemas Evolutivos - MeDSE trilhou um procedimento sistematizado para ficar aderente ao paradigma EPS, baseado em agentes inteligentes mecatrônicos, podendo seguir as normas e, por conseguinte, aderente à Plataforma da Indústria 4.0.

Desde as etapas mais básicas da MeDSE, o conceito de *skill* foi considerado. O problema Global foi gradativamente refinado em problemas regionais e locais, com características atômicas, necessárias para a modelagem dos módulos com a granularidade adequada ao sistema desenvolvido, que no caso do SIAPE optou-se por uma granularidade grossa por definir anagramas completos a serem realizados e não partes desses.

Da modelagem seguiu-se as simulações que garantiram os circuitos e modelos como especificações que foram organizadas e receberam o status de projeto de sistema. O projeto foi realizado seguindo as etapas definidas na MeDSE e o sistema foi transformado em produto. Com a experimentação e as comparações evidenciou-se, entre outros, os parâmetros que classificam o SIAPE como um sistema evolutivo, e com melhor desempenho nas situações que demandam maior agilidade e flexibilidade em tempo real.

Conclusão

Este artigo apresentou os princípios básicos da Metodologia para Desenvolvimento de Sistemas Evolutivos - MeDSE e realizou uma aplicação da metodologia a um sistema denominado de Sistema Inteli-

gente Ágil de Processo Evolutivo - SIAPE, atendendo assim, a necessidade da criação de uma metodologia que contemplasse a noção de agentes inteligentes e suas habilidades, que são denominados de *skills* no paradigma EPS, desde as primeiras fases do desenvolvimento.

Como trabalho futuro espera-se realizar uma experimentação que se possa seguir todos os procedimentos da metodologia MeDSE, primeiramente, para a realização de um SIAPE em escala real para que possíveis melhorias sejam implementadas, incluindo entre outras características, a detecção de falhas. Num segundo projeto, a realização de uma planta industrial aplicada a um produto do parque industrial brasileiro para a realização dos conceitos e operações de sistemas evolutivos, fato que será acompanhado de outros estudos e pesquisas que evidenciarão a eficácia do paradigma EPS na parte real do sistema, e iniciando uma nova fase nas pesquisas de sistemas evolutivos.

Agradecimentos

Aos meus professores e colegas que contribuíram para o desenvolvimento deste trabalho, ao laboratório do CETELI e às agências de fomento SAMSUNG, CNPq, CAPES e FAPEAM.

Referências Bibliográficas

- Abdelaziz, T., Elammari, M., Unland, R. and Branki, C. (2010). Masd: Multi-agent systems development methodology, Multiagent Grid Syst.
- Abele, E., Meyer, T., N'aher, U., Strube, G. and Sykes, R. (2008). Global Production: A Handbook for Strategy and Implementation, Springer Berlin Heidelberg.
- Akillioglu, H. and Onori, M. (2011). Evolvable production systems and impacts on production planning, Proceedings - 2011 IEEE International Symposium on Assembly and Manufacturing, ISAM 2011 (June 2014).
- B. Krogh, E. Lee, I. L. a. M. and Rajkumar, R. (2008). Cyber-Physical Systems, Executive Summary,"Sci. Technol.
- Barbosa, J. and Leitão, P. (2010). Modelling and Simulating Self-Organizing Agent-based Manufacturing Systems.
- Bitkom, V. . Z. (2015). Industrie-40-Whitepaper FuE-Themen, Plattform-i40.de.
- Caire (2001). Technical Information MESSAGE?: Methodology for Engineering Systems of Software Agents, EURESCOM.
- Christensen, J. H. (1994). Holonic Manufacturing System: Initial architecture and standards directions, Presented at First European Conference on Holonic Manufacturing Systems, Hannover, Germany, 1 December 1994., no. March, pp. 1–20.
- Corporation, R. S. (1997). Unified Modelling Language (UML) version 1.0., Rational Software Corporation.
- D. Rotondi, S. Piccione, G. A. A. M. H. J. G. C. K. M. and Trsek, H. (2013). Plug and Work IoT Requirement Assessment and Architecture, IoT Work WP 1, D1.3 pp. 1-135.
- Drath, R. and Horsch, A. (2014). Industrie 4.0: Hit or hype, IEEE Industrial Electronics Magazine.
- Ferreira, P., Doltsinis, S. and Lohse, N. (2014). Symbiotic assembly systems: A new paradigm, Procedia CIRP, pp 26-31.
- Frei, R. M., Ribeiro, L., Barata, J. and Semere, D. (2007). A complexity theory approach to evolvable production systems, International Conference in informatics and control, automation and robotics, Angers, France.
- Hoda A. ElMaraghy. H (2006). Flexible and reconfigurable manufacturing systems paradigms, Flex. Serv. Manuf. J., vol. 17, pp. 261-276.
- Larman, C. (2007). Utilizando UML e padrões: Uma introdução à análise e ao projeto orientado a objetos e ao desenvolvimento iterativo.
- Meyer, B. (2000). Object-Oriented Software Construction, Nucl. Instruments Methods Phys.
- Onori, M. (2002). Evolvable Assembly Systems – A New Paradigm?, Proc.33rd Int. Symp. Robot., pp. 617-622.
- Onori, M.; Barata, J.; Frei, R. (2006). Evolvable assembly systems basic principles, Inf. Technol. Balanc. Manuf. Syst., vol. 220, pp. 1–12.
- Rocha, A., Ribeiro, L. and Barata, J. (2014). A Multi Agent Architecture to Support Selforganizing Material Handling, Technological Innovation for Collective.
- Rosa, R. (2013). Assessing self-organization and emergence in Evolvable Assembly Systems (EAS).
- Rumbaugh, J. (1991). Object-Oriented Modeling and Design.
- Scott A. Deloach, M. F. W. and h. Sparkman, C. (2001). Introduction Software Engineering Knowledge, Engineering. Knowledge, vol. 11, no. 3, pp. 231-258, 2001.
- Ueda, K. (1992). A Concept for Bionic Manufacturing Systems Based on DNAtype Information, PROLAMAT Conference, Man in CIM. Tokyo, Japan, 24–26 June.
- Wooldridge, M. (2002). An Introduction to Multi-Agent Systems, John Wiley and Sons.
- Yoram, K., U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy, H. Van Brussel. (1999). Reconfigurable Manufacturing Systems, "Reconfigurable Manufacturing Systems", vol. 48, pp. 527-540.