

Um estudo da aplicação de Enterprise Information Systems Patterns à modelagem de sistemas de automação

Everton Alves Miranda
eamirand@iff.edu.br
IFF

Jean Felipe Dias de Melo
jean.melo@iff.edu.br
IFF

Rodrigo Andrade Stellet
rastellet@gmail.com
IFF

Aline Pires Vieira de Vasconcelos
apires@iff.edu.br
IFF

Rogério Atem Carvalho
ratem@iff.edu.br
IFF

Resumo: O Enterprise Information Systems Pattern Framework (EIS Pattern Framework) é uma ferramenta didática para modelagem de sistemas, suporte ao desenvolvimento e teste de códigos reutilizáveis, visando principalmente o desenvolvimento de sistemas de Enterprise Resource Planning (ERP). O mesmo evoluiu das experiências obtidas no desenvolvimento do FOS-ERP (Free/Open Source ERP System) denominado ERP5. Tentativas de aplicá-lo a modelagem de sistemas pertencentes a outro domínio não são conhecidas. Este trabalho visa analisar a possibilidade de extensão do uso do EIS Pattern Framework como ferramenta para a modelagem de sistemas automatizados industriais. A metodologia utilizada consistiu da pesquisa bibliográfica sobre aplicações EIS genéricas (incluindo os métodos de modelagem utilizados e a sua correlação com sistemas automatizados) e sobre os detalhes específicos do EIS Pattern Framework (abrangendo seus aspectos históricos e técnicos). A partir disso, foram efetuados testes de aplicação de modelagem via BPMN e via EIS Pattern Framework sobre um mesmo sistema de automação, culminando em uma avaliação comparativa para a referida aplicação. Partindo do entendimento de que as vantagens de um novo modelo devem ir além de sua facilidade de entendimento e de construção, a abordagem avaliou também o desenvolvimento de facilidades para a geração de código. Ao final da análise, chegou-se ao entendimento de que a proposta não apresenta uma relação custo x benefício atraente para aplicação no domínio dos sistemas de automação industrial.

Palavras Chave: EIS Pattern - ERP5 - Ontologia - Elicitação - Sistemas Automáticos

1. INTRODUÇÃO

As técnicas de modelagem são capazes de representar as características de um sistema (e em alguns casos, também o seu comportamento), permitindo, tanto o registro da condição estrutural atual, quanto de suas condições históricas (possivelmente alteradas por sucessivas atualizações), assim como, possibilitando também o registro das condições estruturais esperadas nas atualizações futuras. A representação destas últimas é especialmente útil devido a permitir uniformização de seu entendimento entre os diferentes *stakeholders* e desenvolvedores envolvidos em um projeto (seja ele de criação ou evolução de um sistema), permitindo inclusive documentar este entendimento de forma vinculada a um escopo de contratação de serviços para o aprimoramento do software. Outra importante vantagem da utilização de uma modelagem adequada está na possibilidade de efetuar simulações de comportamento e testes de desempenho, colaborando na identificação precoce de falhas ou inconsistências, minimizando assim, custos de correção.

O *Enterprise Information Systems Pattern Framework (EIS Pattern Framework)* é uma ferramenta didática para modelagem de sistemas, suporte ao desenvolvimento e teste de códigos reutilizáveis, visando principalmente o desenvolvimento de sistemas de *Enterprise Resource Planning (ERP)*.

O objetivo deste trabalho é analisar a possibilidade de extensão do uso do *EIS Pattern Framework* como ferramenta para a modelagem, não somente, de sistemas do ambiente empresarial, mas também, de ambientes relacionados aos sistemas automatizados industriais. A problemática a ser testada é: O *EIS Pattern Framework* pode mostrar-se tão eficiente para modelagem de processos em sistemas de automação quanto é para processos de negócios empresariais? A hipótese inicial é que a resposta seja afirmativa.

Este artigo está estruturado da seguinte forma: Além desta introdução, existem mais quatro seções. A segunda Seção trata da fundamentação teórica sobre modelagem, sendo dividida em 2 subseções: A primeira versa sobre métodos de modelagem voltados para aplicações gerais, enquanto a segunda trata de métodos de modelagem para sistemas de informação; A próxima Seção explica a metodologia de pesquisa utilizada; A quarta Seção aborda o estudo de caso realizado, o qual trata-se de um sistema de automação, possuindo 3 subseções: a primeira aborda o mapeamento através de métodos consagrados, a segunda trata especificamente dos detalhes sobre a modelagem via *EIS Pattern Framework*, enquanto a terceira Subseção demonstra o mapeamento do sistema proposto, via a referida metodologia; Por fim, a quinta Seção refere-se às conclusões.

2. FUNDAMENTAÇÃO TEÓRICA

2.1. MÉTODOS DE MODELAGEM PARA USO GERAL

Existem diferentes métodos para modelagem de sistemas, possuindo cada um o seu conjunto particular de vantagens e desvantagens. Os métodos mais modernos, obviamente fornecem características de qualidade mais evoluídas, porém, o tempo necessário para o desenvolvimento do modelo, assim como, o seu custo, continuam sendo as principais limitações para a ampliação do uso de modelagem. Todavia, é importante ressaltar que o tamanho destas limitações está diretamente associado à amplitude das vantagens e funcionalidades que cada tipo de modelagem pode disponibilizar.

O desenho em escala foi um dos primeiros métodos de modelagem a ser aplicado. Seu emprego é essencial em muitos ramos da engenharia (como por exemplo: nos das áreas de mecânica, civil, elétrica e eletrônica), assim como, nas áreas de arquitetura e agrimensura.

Este tipo de modelagem, inicialmente disponibilizada em formato 2D, permite, àqueles com domínio de suas técnicas, registrar, expressar e transmitir idéias, fornecendo efetividade a comunicação e servindo, após a adequada aprovação dos modelos, como documentos que farão parte do ciclo de vida do componente, sistema ou equipamento.

Com o avanço da informatização, os métodos de modelagens, ditos clássicos, evoluíram consideravelmente, reduzindo o tempo de confecção, aprimorando características e, principalmente, reduzindo o custo destes modelos iniciais. Obviamente, quando se opta por agregar novas funcionalidades a um tipo de modelagem, há algum incremento de custo. Por isso, de forma análoga à realidade de qualquer decisão empresarial, a definição pela utilização (ou não) e, principalmente, a seleção da melhor metodologia de execução da modelagem devem obrigatoriamente passar por uma análise de custo x benefício, de forma a garantir a viabilidade técnica-econômica e o retorno do investimento.

Os avanços obtidos podem ser evidenciados, por exemplo, nas aplicações do tipo CAD (*Computer Aided Design*), as quais permitem desenvolver modelos tanto em formato 2D, quanto em 3D, adicionando inclusive, opções de deslocamento de ponto de vista (movimentação de câmera), tornando possível criar maquetes eletrônicas das instalações (Figura 1). A evolução citada agregou grande facilidade de “leitura” e entendimento do modelo, permitindo um enorme incremento na interação com uma clientela que não dominava as convenções e técnicas utilizadas na representação pelos métodos de desenho mais tradicionais.

Atualmente, esta tecnologia tem obtido um especial destaque na modelagem de sistemas industriais complexos, pois além de identificar com clareza qualquer possível interferência física (por exemplo, as entre tubos, acessórios, equipamentos e elementos estruturais em uma planta de processo), tem sido utilizada com muito sucesso, tanto na identificação de falhas ergonômicas (como por exemplo: dificuldade ou impossibilidade de acesso/visualização de medidores, válvulas e equipamentos), quanto na verificação do espaçamento necessário para desmontagem e retirada de equipamentos durante as campanhas de manutenção, seja ela de ordem preventiva ou corretiva.



Figura 1: Maquete eletrônica de planta de processo

Outro método de grande importante é o de modelagem matemática (Figura 2), o qual já é consagrado no desenvolvimento de sistemas de controle, mediante à simulações de comportamento dinâmico e à previsões de desempenho para os sistemas.

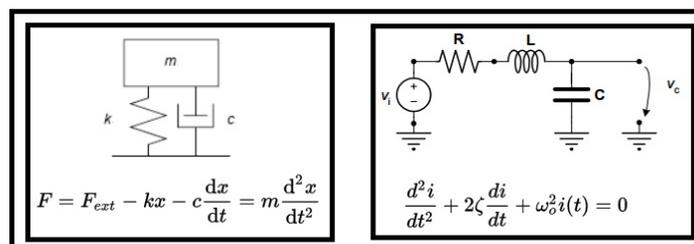


Figura 2: Modelo matemático de sistema mecânico e elétrico

No que tange aos sistemas de automação industrial, existe uma quantidade considerável de diferentes desenhos, diagramas, fluxogramas e outras representações que conjuntamente, modelam um sistema. Diagramas como os exemplificados na Figura 3 são especificamente relacionados a determinados tipos de sistemas automáticos. Neste exemplo se podem observar três sistemas com comportamento parecido, sendo um pneumático (na parte superior esquerda), um hidráulico (à direita) e um eletropneumático (parte inferior). A exemplo do que acontece na área de eletrônica, o projeto destes tipos de sistemas torna-se consideravelmente facilitado pela utilização de seus modelos em softwares de simulação.

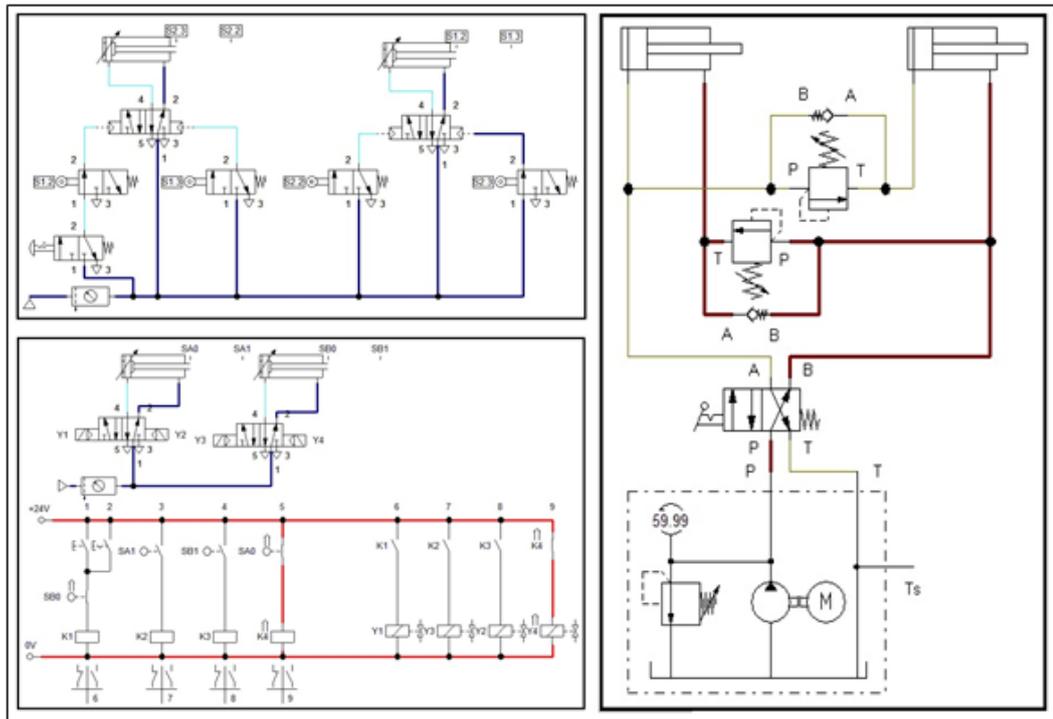


Figura 3: Diagramas Pneumático, Hidráulico e Eletropneumático

Outra forma bastante utilizada para modelar o comportamento de um sistema é através de blocos lógicos (Figura 4) podendo-se modelar sistemas discretos ou contínuos.

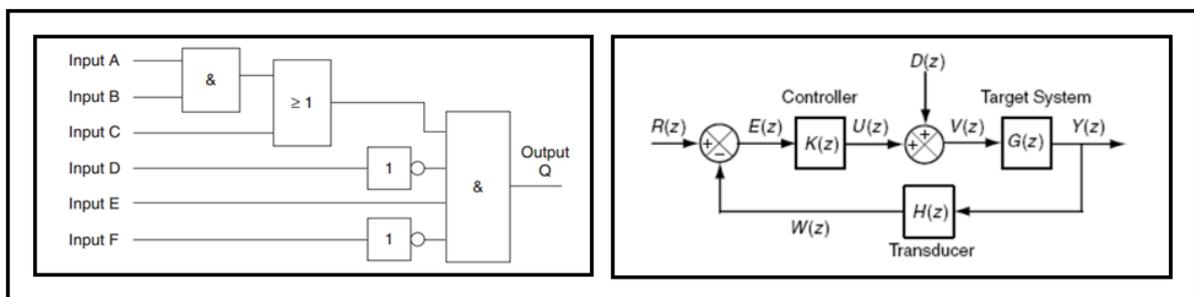


Figura 4: Modelagem por blocos lógicos

Dentre os documentos (modelos) mais utilizados para o registro e entendimento do comportamento automático de um sistema, o “diagrama lógico” possui destaque especial, por ser amplamente utilizado na programação de Controladores Lógicos Programáveis (PLC), os quais são equipamentos microprocessados dedicados à automatização e controle de sistemas. Este tipo de modelo (Figura 5) é confeccionado utilizando-se de blocos lógicos adaptados de forma a exprimir os significados adicionais necessários.

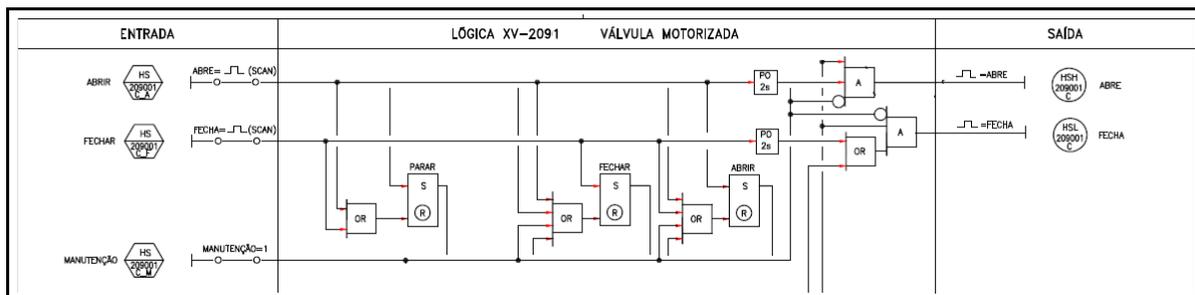


Figura 5: Diagrama Lógico

Conforme a norma IEC 61131-3, os citados PLC podem ser programados através de seis linguagens diferentes. A linguagem de Diagramas de Blocos Funcionais (*Function Block Diagram – FBD*) é parecida com a representação do diagrama lógico, e por isso, apresenta considerável facilidade para a codificação das informações documentadas em forma de software. A programação em diagrama *Ladder* (LD), por sua vez, se assemelha muito à representação utilizada no modelo do projeto elétrico do sistema, fornecendo também facilidades para a codificação destes tipos de sistema.

Outro bom exemplo de linguagem de programação capaz de facilitar a codificação é a realizada por Funções Gráficas de Sequenciamento (*Sequential Function Chart – SFC*) a qual, por ser derivada do Diagrama de Petri, também guarda grande similaridade com a representação (modelo) do sistema. (Figura 6)

Dentre as outras linguagens padronizadas ainda existem duas linguagens textuais (Texto Estruturado – ST e Lista de Instruções – IL).

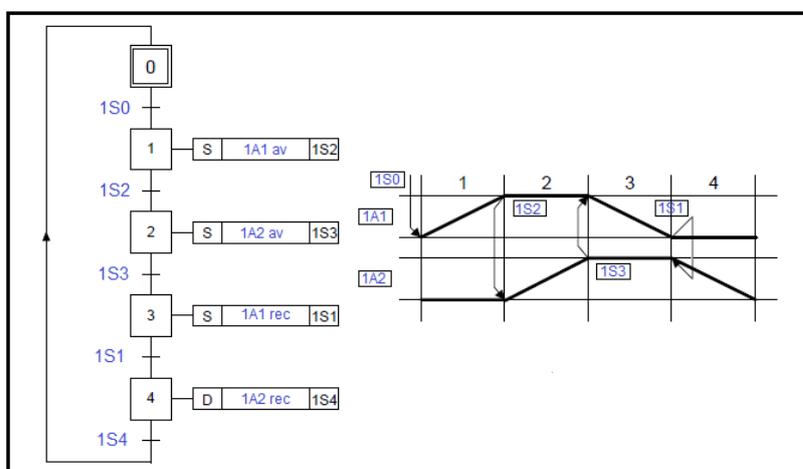


Figura 6: Programa em *Sequential Function Chart*

2.2. MÉTODOS TRADICIONAIS DE MODELAGEM PARA SISTEMAS DE INFORMAÇÃO

Quando se representa sistemas informatizados, um dos métodos mais utilizados é a modelagem através do mapeamento dos processos de negócio envolvidos. Este método busca identificar, tanto os requisitos (funcionais e não funcionais), quanto as restrições, diretas e indiretas do domínio do projeto.

Conforme já exposto em Oliveira (2013), os requisitos funcionais descrevem o que o produto deve fazer, ou seja, que ações processuais ele deve tomar, enquanto os requisitos não funcionais representam as propriedades que o produto deve possuir para desempenhar adequadamente as funções requeridas. O conceito de restrições de projeto refere-se às limitações sobre a especificação do produto, as quais são derivadas da relação do produto com

seu entorno; as diretivas de projeto são as forças associadas ao negócio e o domínio do projeto representa as condições nas quais o projeto deverá ser executado.

Os recursos disponibilizados pela BPMN (*Business Process Model and Notation*) têm facilitado consideravelmente a representação e entendimento do comportamento do sistema, mostrando-se mais eficientes do que os pertencentes ao diagrama de atividades da UML (*Unified Modeling Language*), o qual não possui tantos recursos para esse tipo de modelagem. Nos sistemas em fase de desenvolvimento, é comum realizar-se o mapeamento do estado atual do sistema (*AS IS*) e do sistema que se espera obter após a intervenção (*TO BE*). Apesar das vantagens apresentadas pelo BPMN, mantem-se a utilização da UML, devido às facilidades disponibilizadas pelos diagramas de casos de uso (*USE CASE*) e de classes.

3. METODOLOGIA DA PESQUISA

A pesquisa em curso aborda o detalhamento do *EIS Pattern Framework* e de sua aplicação em paralelo à aplicação de outros métodos de modelagem, sobre um sistema de automação industrial específico, objetivando emitir um parecer quanto a sua eficiência comparada a de outros métodos.

A metodologia iniciou-se com uma pesquisa bibliográfica sobre o estado da arte das aplicações de EIS, os métodos de modelagem utilizados nas mesmas e a sua correlação com sistemas automatizados. Em paralelo foi realizada uma pesquisa sobre os *detalhes* do *EIS Pattern Framework*, abrangendo seus aspectos históricos e técnicos. A próxima etapa foi a aplicação das modelagens baseadas em BPMN ao sistema proposto. Na terceira etapa, foram efetuados testes de aplicação do *EIS Pattern Framework* ao mesmo sistema, culminando em uma avaliação de custo x benefício da referida aplicação.

No início do desenvolvimento do trabalho foi realizado um levantamento de informações sobre o funcionamento do sistema utilizado no estudo de caso, e a partir destas informações, utilizando-se de BPMN, foi desenvolvido o mapeamento “*AS IS*”, o qual, por sua vez, serviu de base para a realização do processo de elicitação dos requisitos e para o desenvolvimento do modelo “*TO BE*”.

Após a pesquisa bibliográfica e a utilização de BPMN para mapeamento e registro do modelo de negócio, buscou-se modelar o sistema e avaliar os resultados desta modelagem quanto à facilidade de construção e de entendimento, assim como, quanto a possibilidade de agregar facilidades para a geração de código.

Por fim, foi realizada a análise comparativa dos modelos.

3.1. DETALHAMENTO DA PESQUISA BIBLIOGRÁFICA

Para a pesquisa sobre o estado da arte, foi escolhida a base *SCOPUS* e definida uma *string* de busca construída a partir das palavras e termos entendidos como chaves para a representação dos objetivos. Na primeira tentativa foi utilizada uma *string* composta por três termos, obtendo-se um resultado de 82 fontes de pesquisa, porém buscando-se um maior refinamento, utilizou-se uma nova *string* que recebeu um quarto termo e retornou 21 fontes.

A Figura 7 apresenta a quantidade de resultados obtidos na busca individual e em cada combinação das palavras chaves utilizadas na primeira tentativa, assim como o resultado consolidado da segunda tentativa.

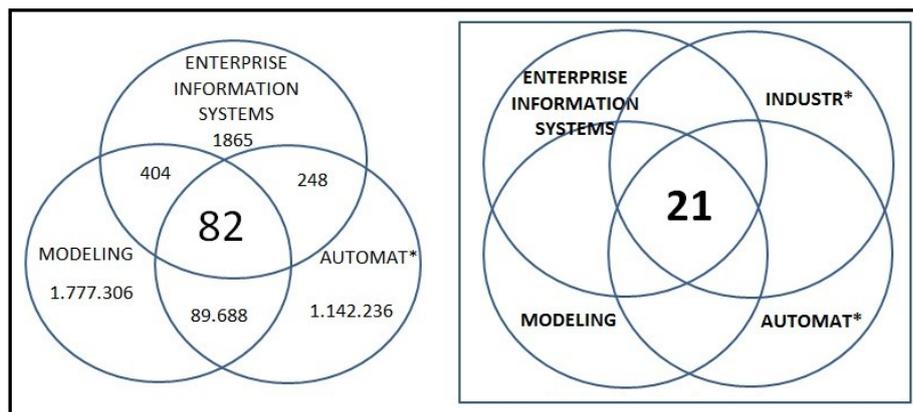


Figura 7: Resultados obtidos por combinação de filtros

Partindo desse segundo resultado foram definidos os critérios de priorização e exclusão privilegiando os artigos com maior número de citações e excluindo aqueles nos quais o *Abstract* não demonstrava adequada aderência aos objetivos. Assim, o resultado convergiu para um total de 6 artigos, os quais foram analisados de forma integral (vide Tabela 1).

Tabela 1: Fontes de pesquisa selecionadas após aplicação dos critérios de priorização / exclusão

Nº	Citações	Ano	Título
1	74	2008	The importance of business process modeling in software systems design
2	20	2013	An ontology-driven framework towards building enterprise semantic information layer
3	8	2012	Semantics enactment for interoperability assessment in enterprise information systems
4	7	2013	Agent-based workflow approach to the design and development of cross-enterprise information systems
5	7	2012	From a high level business process model to service model artifacts: A model-driven approach
6	0	2017	Recalling the rationale of change from process model revision comparison – A change-pattern based approach

Quanto à pesquisa histórica e técnica sobre os detalhes do *EIS Pattern Framework*, a qual demanda um vínculo maior com o conhecimento que já está consolidado, do que com o ainda em construção, foram priorizadas as referências indicadas por um especialista da área.

4. O ESTUDO DE CASO

4.1. CARACTERÍSTICAS DO SISTEMA MODELADO

O sistema objeto do estudo visa o monitoramento de dados ambientais em ambientes lagunares, mais especificamente, o monitoramento do volume fitoplânctônico, o qual será determinado através da medição da foto-fluorescência.

A escolha do método de medição baseou-se em dois motivos: o primeiro e que, no fitoplâncton existem diversos pigmentos fotossintéticos (Ex: as *clorofilas a, b e c*, os carotenos, as xantofilas, entre outros), sendo a *clorofila a*, o principal pigmento fotossintético de todos os organismos que realizam fotossíntese com liberação de oxigênio. Devido a isso,

ela tornou-se amplamente utilizada para estimar a biomassa fitoplanctónica em águas doces superficiais (INAG, 2009); O segundo motivo é que, a medição da fotofluorescência, mediante a sensibilização por LED com o adequado comprimento de onda, apresenta-se como um método adequado e relativamente barato para a interpretação do volume de clorofila a em corpos de água bruta (PUIU et al., 2015). O trabalho de Leeuw, Boss e Wright (2013) também endossa essa segunda motivação.

Na atualidade, o processo do referido sistema é realizado, quase que integralmente, de forma manual, mas, a partir deste momento, este será totalmente automatizado.

Utilizando-se do método de entrevista foi realizado o levantamento do funcionamento atual do sistema, o qual foi representado no diagrama “AS IS” conforme a Figura 8:

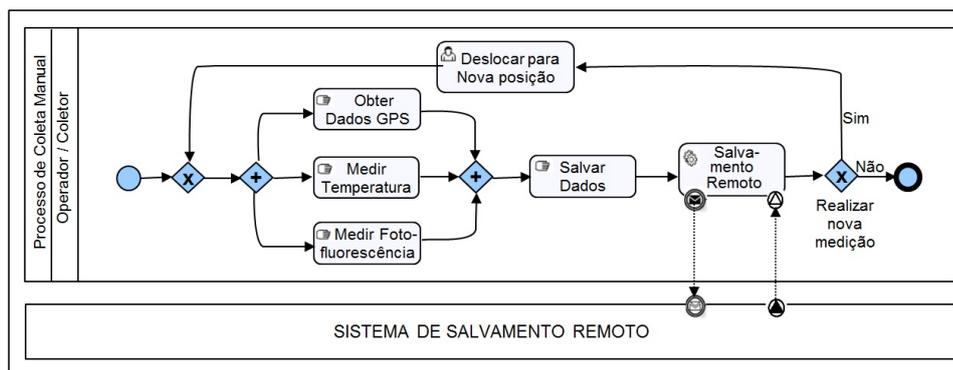


Figura 8: Mapeamento “AS IS”

Partindo desse primeiro diagrama, mediante ao uso da técnica de *Brainstorming*, foi concluída a elicitação dos requisitos do sistema, resultando nos seguintes requisitos funcionais: reconhecimento do posicionamento do sistema, data e hora da medição (através do GPS); medição da temperatura; medição da foto-fluorescência; armazenamento local dos dados e envio dos dados para armazenamento remoto.

Foram identificados também: um requisito não-funcional relativo a confiabilidade que, por sua vez, motivou o requisito funcional de salvamento remoto; e uma restrição de projeto, a qual exige um baixo custo de construção para cada unidade a ser fabricada.

Após as investigações de soluções, baseadas nos requisitos e restrições elicitados, foi decidido que o sistema será composto por uma unidade de processamento, um sensor de foto-fluorescência, um sensor de temperatura, um módulo de GPS, uma unidade de leitura e gravação de cartão SD e uma antena de radiofrequência. O modelo “TO BE” pode ser visto na Figura 9.

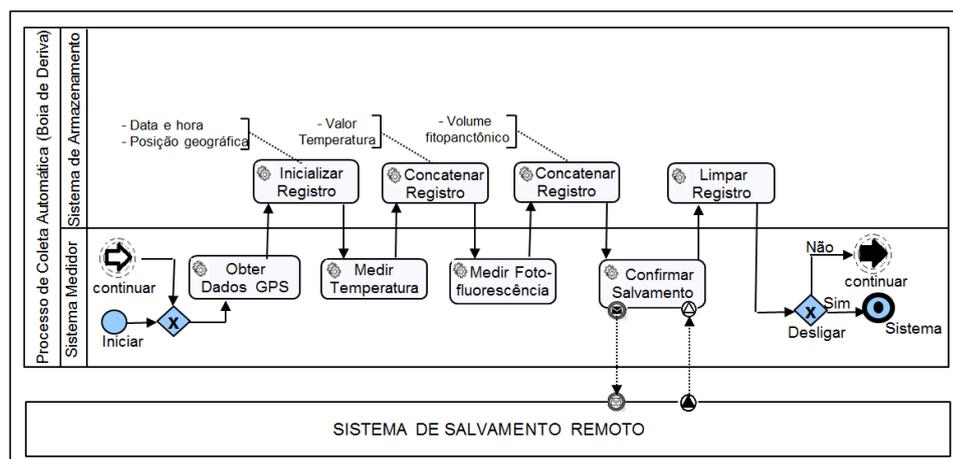


Figura 9: Mapeamento “TO BE” do sistema a ser implementado

Todo o sistema ficará embarcado em uma boia de deriva, a qual terá liberdade para deslocar-se livremente na superfície da lagoa, realizando medições em pontos aleatórios, sendo o intervalo entre as amostragens definido na configuração do sistema.

O processo de funcionamento inicia-se com a leitura do posicionamento via GPS e inicialização de um registro, o qual conterá também a data e a hora da mesma. A próxima etapa é a medição da temperatura da água e a concatenação do resultado ao registro já inicializado. A terceira medição é a relativa à foto-fluorescência, sendo também concatenada ao registro.

Após a montagem do registro, o mesmo deve ser gravado em um cartão SD e, logo após, enviado via antena de RF para o sistema de armazenamento remoto.

Concluído o envio haverá uma confirmação do salvamento remoto, porém, independentemente da recepção da confirmação, o sistema deve prosseguir com sua sequência, destruindo o registro temporário e reiniciando o ciclo de medição após a contagem do tempo configurado entre medições.

4.2. METODOLOGIA DE MODELAGEM VIA *EIS PATTERN FRAMEWORK*

Um *framework* tem como proposta direcionar o processo de modelagem a partir das especificações (requisitos funcionais, requisitos não funcionais, processos de negócio e restrições) e, após a adequada análise, transformar esses dados em especificações formais de sistemas de informação (OLIVEIRA, 2013).

O *EIS Pattern Framework* evoluiu das experiências obtidas no desenvolvimento do ERP5 que, por sua vez, se caracteriza como um ERP Livre de Código Aberto (*Free/Open Source ERP System – FOS-ERP*) baseado em um modelo de ontologia unificado que possui como núcleo, cinco entidades de negócios. São elas, *Recurso, Nó, Caminho, Movimento e Item*.

Conforme Carvalho e Monnerat (2007), o ERP5 foi concebido para ser um *framework* muito flexível para o desenvolvimento de aplicações empresariais, e neste sentido, deve ser suficientemente abstrato para abranger todos os conceitos básicos de negócios, sendo adaptável a vários modelos sem incorrer em altos custos de mudanças e manutenção.

O ERP5 foi desenvolvido por um grupo de empresas e instituições de ensino e pesquisa, envolvendo países como França e Brasil, sendo um projeto de código aberto que tem por objetivo oferecer soluções de alta tecnologia e custo acessível para as empresas interessadas em adotar um sistema integrado de gestão (SANTOS et al., 2010).

O referido *framework* representa uma abordagem simplificada que, baseada em modelos, fluxos de trabalho simplificados e alto nível de reutilização, é capaz de reduzir não apenas o esforço de programação, mas também o número e a complexidade das tarefas de modelagem, aumentando a produtividade, facilitando o gerenciamento e minimizando os erros de modelagem e codificação a partir do uso intensivo de ferramentas altamente integradas (MONNERAT; DE CARVALHO; DE CAMPOS, 2008).

O *EIS Patterns* consiste em um *framework* simplificado, focado em testar novas técnicas para desenvolver EIS Flexíveis, e sua concepção teve como inspiração os conjuntos de Lego, ou seja, blocos de construção básicos que podem ser combinados com entidades diferentes. O mesmo foi construído em torno de três conceitos abstratos essenciais (*Recurso, Nó e Movimento*), cada um com três subclasses, as quais representam dois conceitos "opostos" derivados e um terceiro que funciona como agregador destes dois primeiros (CARVALHO; JOHANSSON, 2013).

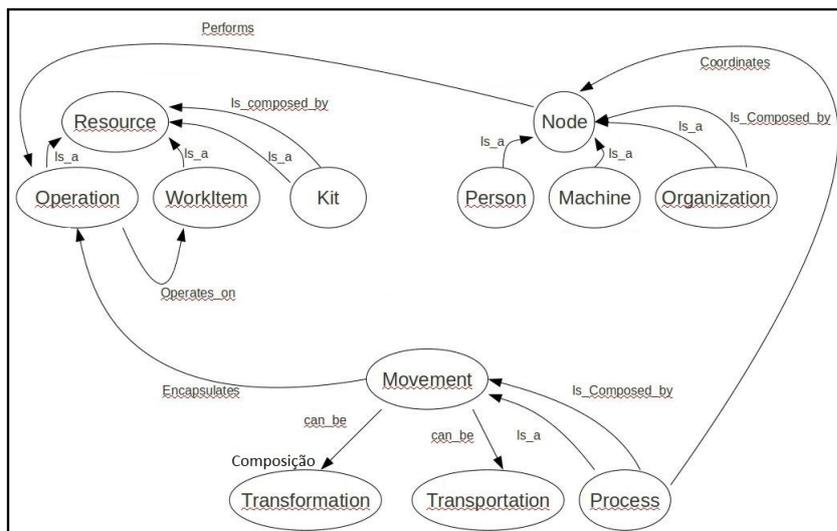


Figura 10: Ontologia representativa do núcleo do *EIS Pattern* (CARVALHO; JOHANSSON, 2013)

A ontologia do *EIS Patterns Framework* foi representada conforme a Figura 10 e seus conceitos são interpretados conforme a lista abaixo:

- **Recurso:** é tudo o que é usado para a produção. Tem um papel passivo, sendo responsável por armazenar dados de recursos de produção. Pode ser caracterizado como:
 - Material: produto, componente, ferramenta, documento, matéria-prima, etc;
 - Operação: operação humana ou de máquina, bem como seus derivados;
 - Kit: é um conjunto de recursos materiais e/ou imateriais. Ex.: pacotes de serviços e componentes para fabricação.
- **Nó:** é algo que transforma recursos. Tem um papel ativo, sendo responsável pela execução das operações de produção. Pode ser caracterizado como:
 - Pessoa: funcionário, pessoa de contato do fornecedor, operador de perfuração, etc;
 - Máquina: hardware, software, equipamento de perfuração, conta bancária, etc;
 - Organização: é um coletivo de máquinas e/ou pessoas. Ex.: célula de fabricação, departamento, empresa, governo, etc.
- **Movimento:** é uma transferência de um recurso entre dois nós. Tem uma função de coordenador, sendo responsável por gerenciar os nós enquanto trabalham em recursos. Pode ser caracterizado como:
 - Transformação: é um movimento dentro de um nó, ou seja, a fonte e o destino são o próprio nó; representa a transformação por máquina ou por trabalho de um recurso. Ex.: perfurar uma placa de metal ou escrever um relatório.
 - Transporte: é um movimento de recursos entre dois nós distintos. Ex.: transferência de um componente de uma estação de trabalho para outra ou envio uma ordem do fornecedor para o cliente.
 - Processo: é um conjunto de transformações e/ou transportes, ou seja, um processo de negócio.

O conceito *Nó* pode ser estendido através do padrão *Decorator*. A extensão do conceito *Recurso* dá-se através de subclasses, enquanto o conceito *Movimentos* é estendido através da configuração.

A Ontologia demonstra uma cadeia de relacionamentos que denota como o núcleo implementa os processos de negócios: "um *Processo* coordena o(s) *Nó(s)* para executar a(s) *Operação(ões)* que, por sua vez, opera(m) em *Item(ns) de Trabalho*". O significado semântico

dessa cadeia é que os objetos de *Processo* controlam, sob determinadas condições, objetos de *Nó* que realizam operações para transformar ou transportar *Recursos*. Isto leva a outra relação especial que é: "um *Movimento* encapsula uma *Operação*", o que significa que um objeto de *Movimento* encapsulará a execução de uma *Operação*. Em termos práticos, uma *Operação* é a descrição abstrata de uma operação de produção, que é implementada através um ou mais métodos de objetos de *Nó*. Quando essa *Operação* é disparada por um objeto de *Processo*, ela difere a execução real para um método de objeto de *Nó* pré-configurado e essa execução é registrada por um objeto de *Movimento*, que armazena todos os parâmetros, como por exemplo, data, hora e resultados dessa execução. Portanto, uma *Operação* é um conceito abstrato que pode ser configurado para atribuir uma execução diferenciada, para diferentes métodos ou de diferentes objetos, de acordo com as intenções de cada instância de processo de negócios específico. Em outras palavras, uma abstração de processo de negócios mantém sua lógica, enquanto resultados específicos podem ser obtidos mediante a configuração (CARVALHO; JOHANSSON, 2013).

Esse mecanismo permite que um determinado modelo de processo de negócios possa ser configurado para ser implementado de diferentes maneiras (de acordo com diferentes contextos), para diferentes aplicações, até mesmo em tempo de execução.

É importante notar que neste ambiente, os *Processos* controlam os elementos ativos, os *Nós*, que por sua vez operam em cima dos elementos passivos, os *Recursos*.

Em termos de programação, isso significa que os *Processos* são configuráveis, os *Nós* são estendidos e os *Recursos* são tipicamente classes do tipo "data bag".

4.3. A NOVA MODELAGEM DO SISTEMA

A modelagem é um pré-requisito para permitir a compreensão comum do sistema, em suas várias interações, a fim de "fornecer as informações certas, no momento certo, no lugar certo". No entanto, é comum observar-se problemas oriundos da falta de compreensão da semântica dos modelos elaborados (YAHIA et al., 2012).

Conforme já expresso em Barjis (2008), se um modelo consegue capturar o fluxo do processo, todas as atividades principais, iniciadores e executores de cada atividade, ordem pontual de atividades e resultados criados (saída) por cada atividade, ele aumenta consideravelmente a probabilidade da obtenção de um projeto adequado.

Não é suficiente apenas conceber as atividades de negócios ligadas por fluxos de controle do processo. Para representar a totalidade dos requisitos, uma definição de processo deve indicar explicitamente todas as entidades que participam no processo. Esses requisitos devem ser transformados, sem perda de informações, em especificações semânticas, das quais, diferentes componentes de software podem ser derivados (SOLTANI; BENSLIMANE, 2012).

A análise da bibliografia demonstrou que as pesquisas estão muito focadas em como automatizar a geração de códigos a partir de modelagem baseada em alguma ontologia específica. Os trabalhos de Barjis (2008); Soltani e Benslimane (2012); Wang, Shen e Hao (2006) e Yahia et al. (2012) demonstram diferentes propostas que, de alguma forma, almejam este objetivo. Os estudos de Wang; Shen; Hao (2006) abordam também a interoperabilidade entre diferentes sistemas, assim como, o trabalho apresentado por Song, Zacharewicz e Chen (2013). Não se encontrou pesquisas relevantes sobre a aplicação específica de ontologias à modelagem de sistemas de automação.

Mediante ao exposto, foi realizado um teste da modelagem através da ontologia *EIS Partten*, partindo da total ausência de referências anteriores. O resultado pode ser observado na Tabela 2 e na Figura 11.

Figura 11: Nova Modelagem do Sistema

A metodologia mais usual de programação, oriunda de modelos baseados em UML, costuma estender um comportamento de uma classe a partir da criação de subclasses da mesma, porém, esta técnica pode conduzir à hierarquias de classe complexas e consideravelmente difíceis de manter.

Conforme expresso em Carvalho e Johansson (2013), enquanto o uso de subclasses adiciona um comportamento a todas as instâncias da classe original, uma solução utilizando o padrão *Decorator* pode fornecer novo comportamento, em tempo de execução, para objetos individuais. A utilização de *Decorator* facilita a configuração do sistema, no entanto, normalmente, é necessário lidar com muitos pequenos objetos.

A utilização de decoradores permite, de acordo com a necessidade da lógica do processo, associar e/ou dissociar diferentes responsabilidades a objetos de *Nó*, permitindo que: o mesmo objeto, com o mesmo identificador, seja utilizado durante todo o processo de negócio, não havendo necessidade de criar diferentes objetos de diferentes classes. Além disso, é possível seguir o mesmo objeto durante todo o seu ciclo de vida, inclusive por meio de diferentes processos de negócios, pois, depois que um objeto é criado e validado, ele manterá sua identidade para sempre. Essa realidade traz facilidade na realização de auditorias.

Os decoradores devem manter um conjunto de regras de associação, o qual é responsável por permitir ou proibir que novas responsabilidades sejam atribuídas aos objetos específicos. Assim, apenas se um determinado objeto respeita as regras de associação de um determinado Decorador, ele poderá ser decorado pelo referido Decorador.

A avaliação realizada tomou como base, entre outras premissas, algumas considerações expressas em Soltani e Benslimane (2012), as quais definem que o desenvolvimento de um aplicativo corporativo em grande escala (por exemplo: aplicações de *Enterprise Information Systems*), sempre começa com a abstração de mais alto nível, nas quais se encontram a especificação e a representação do negócio sob a forma de modelos de processos de negócios. Em paralelo, Carvalho e Campos (2009) expõem que, a maior parte do desenvolvimento e customização de software é feita através de ciclos de vida interativos e incrementais, nos quais não há limites claros entre a fase de definição dos requisitos e as fases de projeto preliminar, tampouco, entre a fase de projeto detalhado e as de implementação.

Devido ao entendimento de que as vantagens de um novo modelo devem ir além de sua facilidade de entendimento e de construção, agregando, principalmente, facilidades para a geração de código. As tentativas de implementação de código foram direcionadas a utilização dos Decoradores, buscando o adequado suporte às abstrações de alto nível, condizentes com o *EIS Pattern Framework*.

O entendimento alcançado foi que, apesar da demonstração da capacidade do *EIS Pattern Framework* gerar modelos representativos para sistemas de automação, a utilização de um novo método de modelagem só se justifica, caso ele consiga trazer vantagens em relação aos métodos com uso já consagrado.

Chegou-se à conclusão de que a modelagem proposta não apresentou vantagens significativas, principalmente pelo reconhecimento de que os conceitos utilizados pelo *EIS Pattern Framework* utiliza um grau de abstração excessivamente elevado, o qual apesar de demonstrar-se eficiente para sistemas compatíveis (como por exemplo: os sistemas de gestão, os financeiros, os de alocação de recursos, os de fluxo de materiais, etc), não se enquadra bem aos sistemas de automação, os quais, por pertencer a um domínio de aplicação muito diferente (sendo compostos por elementos essencialmente concretos), não têm facilidade para

incorporar níveis de abstração tão elevados e, por isso, acabam não sendo favorecidos pelo uso dos decoradores.

Partindo-se de todo o exposto, identificando os benefícios inexpressivos apresentados e prevendo uma geração de custos adicionais (normalmente associados a treinamento da equipe com foco em uma mudança de metodologia ou em uma quebra de paradigma), considerou-se a aplicação do *EIS Pattern Framework* como não recomendado para sistemas de automação.

5. CONCLUSÕES

O *EIS Pattern Framework* evoluiu das experiências obtidas no desenvolvimento de um FOS-ERP (*Free/Open Source ERP System*) denominado ERP5, o qual consiste em um *framework* simplificado, focado em testar novas técnicas para desenvolver EIS Flexíveis. A sua concepção foi baseada na utilização de blocos de construção básicos, os quais podem ser combinados com entidades diferentes, sendo construído em torno de três conceitos abstratos essenciais (*Recurso, Nó e Movimento*), cada um com três subclasses, as quais representam dois conceitos "opostos" derivados e um terceiro que funciona como agregador destes dois primeiros.

O objetivo desta pesquisa foi analisar a possibilidade de extensão do uso do *EIS Pattern Framework* como ferramenta aplicável à modelagem de ambientes relacionados aos sistemas automatizados industriais, avaliado tanto a facilidade de construção e de entendimento do modelo, quanto a possibilidade de favorecimento à geração de código.

Foi percebido que devido ao fato dos sistemas de automação pertencer a um domínio de aplicação muito diferente (sendo compostos por elementos essencialmente concretos), os mesmos não apresentam facilidade para a incorporação de níveis de abstração tão elevados quanto os utilizados no *EIS Pattern Framework*.

Assim, apesar de *EIS Pattern Framework* ter demonstrado capacidade para gerar modelos representativos de sistemas de automação, entendeu-se que o mesmo não traz vantagens expressivas que justifiquem sua aplicação aos referidos sistemas.

6. REFERÊNCIAS BIBLIOGRÁFICAS

BARJIS, J. The importance of business process modeling in software systems design. *Science of Computer Programming*, v. 71, n. 1, p. 73–87, 1 mar. 2008.

CARVALHO, R. A. DE; CAMPOS, R. DE. Uma análise de aspectos relacionados ao desenvolvimento e adoção de Enterprise Resources Planning livre de código aberto. *Gestão & Produção*, v. 16, n. 4, p. 667–678, dez. 2009.

CARVALHO, R. A. DE; JOHANSSON, B. Towards More Flexible Enterprise Information Systems. In: *Enterprise Information Systems of the Future*. Berlin, New York: Springer Berlin Heidelberg, 2013. p. 157–164.

CARVALHO, R. A. DE; MONNERAT, R. M. ERP5: Designing for Maximum Adaptability. In: *Beautiful Code: Leading Programmers Explain How They Think*. First Edition ed. Sebastopol, California: O'Reilly Media, Inc., 2007. p. 339–352.

INAG, I. P. Manual para a avaliação da qualidade biológica da água. Protocolo de amostragem e análise para o Fitoplâncton. Ministério do Ambiente, do Ordenamento do Território e do Desenvolvimento Regional. Instituto da Água, I.P., , jul. 2009. Disponível em: <<https://www.apambiente.pt/dqa/assets/protocolo-de-amostragem-e-an%C3%A1lise-para-o-fitopl%C3%A2ncton.pdf>>. Acesso em: 7 maio. 2017

LEEUEW, T.; BOSS, E. S.; WRIGHT, D. L. In situ Measurements of Phytoplankton Fluorescence Using Low Cost Electronics. *Sensors (Basel, Switzerland)*, v. 13, n. 6, p. 7872–7883, 19 jun. 2013.

MONNERAT, R. M.; DE CARVALHO, R. A.; DE CAMPOS, R. [UNESP. Enterprise systems modeling: The ERP5 development process. Proceedings of the ACM Symposium on Applied Computing, p. 1062–1068, 1 dez. 2008.

OLIVEIRA, V. C. DE. Modelagem e design de sistemas de serviço para automação. text—São Paulo: Universidade de São Paulo, 7 jun. 2013.

PIIU, A. et al. Submersible Spectrofluorometer for Real-Time Sensing of Water Quality. Sensors, v. 15, n. 6, p. 14415–14434, 18 jun. 2015.

SANTOS, R. B. R. et al. Ensino de ERP de código aberto suportado por Jogos de Empresa. INGEPRO – Inovação, Gestão e Produção, v. 2, p. 10–22, Dezembro de 2010.

SOLTANI, M.; BENSLIMANE, S. M. From A High Level Business Process Model To Service Model Artifacts: A Model-Driven Approach. 14-th International Conference on Enterprise Information Systems (ICEIS 2012), jun. 2012.

SONG, F.; ZACHAREWICZ, G.; CHEN, D. An ontology-driven framework towards building enterprise semantic information layer. Advanced Engineering Informatics, Modeling, Extraction, and Transformation of Semantics in Computer Aided Engineering Systems. v. 27, n. 1, p. 38–50, jan. 2013.

WANG, S.; SHEN, W.; HAO, Q. An agent-based Web service workflow model for inter-enterprise collaboration. Expert Systems with Applications, Computer Supported Cooperative Work in Design and Manufacturing. v. 31, n. 4, p. 787–799, nov. 2006.

YAHIA, E. et al. Semantics enactment for interoperability assessment in enterprise information systems. Annual Reviews in Control, v. 36, n. 1, p. 101–117, Abril 2012.