

# Computação Sensível a Contexto Aplicada a Sistemas Elétricos

Jonysberg P. Quintino e Carlos A. G. Ferraz

**Resumo** – Integrar sistemas de supervisão e controle para monitorar o desempenho da rede elétrica, analisando a segurança e a confiabilidade dos equipamentos é tarefa rotineira dos operadores dos centros de controle. No entanto, na maioria dos casos, esta integração não é automática, exigindo a visualização de informações em diferentes telas de sistemas distintos, o entendimento do contexto sobre o estado operativo de equipamentos, e a redigitação de dados em comum entre os sistemas, entre outros. Um estudo analisou dados de uma grande empresa de transmissão de energia elétrica e descobriu um número significativo de cancelamentos de solicitações de intervenção, por exemplo, devido a erros de digitação ou a inconsistências causadas pela falta de informação de contexto. Este trabalho apresenta a arquitetura de uma plataforma de software capaz de inferir contexto e integrar sistemas. Um protótipo funcional foi implementado como prova de conceito, e os resultados preliminares apontam para uma maior eficiência na operação de um centro de controle de última geração.

**Index Terms** – Power Engineering Computing, Middleware, Context-Awareness, SCADA/Energy Management Systems

## I. INTRODUÇÃO

Há décadas, os concessionários em todo o mundo contam com sistemas informatizados de gestão e supervisão para planejar e controlar a geração, transmissão e distribuição de energia [1]. De acordo com Masiello [1], tais sistemas fazem, entre outras coisas, o controle em tempo real de todos os equipamentos de geração de energia e monitoramento do desempenho do sistema de transmissão, ajudando os operadores do sistema a assimilar o que está acontecendo na rede de energia. Ao longo das últimas décadas, muitos programas de computador e aplicações de software foram desenvolvidos para exercer o controle da supervisão e aquisição de dados (*SCADA – Supervisory, Control And Data Acquisition*),

registrar eventos, realizar análises estatísticas, prever o efeito das interrupções para melhorar a segurança, etc. Hoje em dia, em consequência, as concessionárias usam uma grande variedade de sistemas para monitorar e gerenciar seus equipamentos.

Alguns desses sistemas, quando utilizados em conjunto, aumentam potencialmente os seus benefícios. No entanto, estes sistemas não são integrados automaticamente, tornando mais difícil o trabalho dos operadores dos centros de controle. Por exemplo, a fim de identificar o motivo de um desligamento de equipamento, os operadores têm de acessar informações que são exibidas por diferentes sistemas, tendo que redigitar dados de um sistema para outro a fim de consolidar informações sobre o estado de funcionamento de um equipamento, e outras situações que são típicas de um ambiente não integrado.

Sabe-se que grandes volumes de dados são colhidos pelos sistemas de gestão de energia e que estes dados precisam ser transformados em informações significativas/relevantes quanto a um dado contexto para serem apresentadas aos operadores. Ao analisar dados de 2011-2012 da CTEEP (Companhia de Transmissão de Energia Elétrica Paulista), uma das maiores empresas de transmissão de energia do Brasil, identificou-se que 42% dos cancelamentos das solicitações de intervenção para fins de manutenção se deram devido a erros na entrada de dados, especialmente causados durante a redigitação ou pela falta de informações de contexto suficientes. A análise também mostrou que muitos conjuntos de dados provenientes de diferentes sistemas, são especialmente relacionados, podendo formar um contexto [2]. Por exemplo, os dados sobre o identificador de um equipamento programado para manutenção e a data desta manutenção, ambos oriundos do sistema de programação de intervenções, e o registro (naquela data) do desligamento do mesmo equipamento monitorado pelo sistema SCADA, permitem inferir e registrar automaticamente o contexto 'O desligamento não é forçado', entre outros.

Em [3] é apresentada uma visão, originalmente proposta por P. Zhang, F. Li e N. Bhatt [4], de centros de controle do futuro, também chamados de *smart control centers*, onde uma de suas

---

Este trabalho foi desenvolvido no âmbito do Programa de Pesquisa e Desenvolvimento Tecnológico do Setor de Energia Elétrica regulado pela ANEEL.

Os autores gostariam de agradecer a todos os membros da equipe do projeto PD-0068-0022/2012, em particular Marcos Bertinotti, desenvolvido em parceria UFPE-CTEEP-In Forma Software com recursos do Programa de P&D ANEEL.

Jonysberg P. Quintino e Carlos A. G. Ferraz trabalham no Centro de Informática – Cin da Universidade Federal de Pernambuco - UFPE (e-mails {jq,cagf}@cin.ufpe.br).

características-chave é o monitoramento *online* centrado no ser humano. Para tanto, as funções de monitoramento da próxima geração devem fornecer aos operadores informações úteis em vez de apenas dados não processados. Estas funções devem empregar técnicas de visualização com o objetivo de ajudar cada operador a digerir informação rapidamente. Neste artigo, adicionalmente propomos o uso de sensibilidade a contexto ou computação sensível a contexto (definida em III.A) como forma de fornecer ao operador (usuário) informações e/ou serviços relevantes quanto à tarefa a ser realizada.

Visando à integração dos sistemas de software envolvidos nas ações de controle, monitoramento e execução de intervenções no sistema de transmissão de energia elétrica para produzir informações relevantes quanto a tais ações, este trabalho apresenta uma plataforma de software (*middleware*) composta de serviços sensíveis a contexto, capazes de adquirir dados e transformá-los em informações de contexto. Um protótipo foi desenvolvido para integrar um sistema supervisório (SCADA) e outros sistemas que são utilizados para planejar, gerenciar e analisar eventos relacionados à transmissão de energia elétrica. Até o momento da escrita deste artigo, o protótipo permitia a simulação da notificação de bloqueios, desligamentos e normalizações de equipamentos, com base nas informações de contexto.

O restante deste artigo está estruturado da seguinte forma: a seção II apresenta uma análise do problema; a seção III discute o panorama tecnológico envolvido nesta proposta; a seção IV explica a arquitetura proposta para o *middleware* sensível ao contexto; a seção V demonstra um protótipo desenvolvido para prova de conceitos e, por fim, as conclusões são apresentadas na seção VI.

## II. ANÁLISE DO PROBLEMA

Os centros de controle geralmente são divididos em três diferentes áreas de atuação: pré-operação, tempo real e pós-operação. A pré-operação analisa e aprova solicitações de impedimento, por exemplo. A operação em tempo real é responsável por controlar os níveis operacionais (por exemplo, tensão, carga e controle de frequência) da rede de energia elétrica, ao passo que a pós-operação analisa e registra ocorrências, tais como falha de equipamento ou desligamentos de linhas de transmissão.

Sistemas computacionais especializados, como SCADA, permitem o monitoramento do estado operativo de um equipamento em tempo real. Há também sistemas responsáveis pela solicitação, liberação e normalização de impedimentos, como também pelo registro dos fatos

(inicialização/ocorrência/finalização) para uma análise mais profunda dos acontecimentos.

Normalmente, estes sistemas não são integrados, o que significa que solicitações de impedimento (SI) emitidas passam por verificações e intervenções manuais para validar e garantir que as informações contidas nas programações de manutenção, por exemplo, estejam corretas e que ocorram de forma a diminuir ou tornar mais eficiente o tempo de parada. Como exemplo, é possível se aproveitar desse tempo (de parada) para incluir novos serviços em um mesmo equipamento ou em equipamentos que estejam relacionados com uma função de transmissão já autorizada a parar de funcionar. Eventuais erros podem acarretar em cancelamentos de SI, tanto por falta de integração, como também, por falta de informações de contexto que garantam o correto registro das informações dos eventos. A Tabela 1 apresenta um levantamento de SI emitidas e canceladas durante o período de 2011-2012 em uma das maiores empresas de transmissão do país (CTEEP), responsável por mais de 12 mil km de linhas de transmissão.

Tabela 1 Solicitações de Impedimento em 2011-2012. Fonte: CTEEP

Solicitações de impedimento	Qtd.	%
Autorizadas	48.157	36
Canceladas	87.395	64
<b>Total</b>	<b>135.552</b>	<b>100</b>

Do total de solicitações emitidas, observa-se que 87.395 (64%) foram canceladas. A análise detalhada sobre os possíveis motivos de cancelamento mostrou que 36.682 (42% do total de cancelamentos) foram relativos à falta de uma visão integrada de dados provenientes de diferentes sistemas computacionais e/ou à falta de informações contextuais, enquanto que 50.713 (58% dos cancelamentos) se deram por outros motivos. Como o escopo desse trabalho é a integração dos sistemas de software envolvidos nas ações de controle, monitoramento e execução de intervenções no sistema de transmissão de energia elétrica, a análise focou nos cancelamentos de SI relacionados à falta de integração dos sistemas envolvidos, tanto em relação a dados provenientes de diferentes sistemas quanto em relação a informações contextuais (Tabela 2).

Tabela 2 Detalhamento dos motivos de cancelamento de SI.  
Fonte: CTEEP

Ações necessárias (pendentes)	Qtd.
a. Correção no preenchimento	31.360
b. Ajuste de horário (período)	3.212
c. Adequar condições de segurança	1.970
d. Inclusão de serviço dentro do prazo	140
<b>Total</b>	<b>36.682</b>

Dentre os motivos dos cancelamentos, o erro no preenchimento dos dados (a) é o fator que mais contribuiu, principalmente, devido à falta de integração entre os sistemas envolvidos. O ajuste de horário (b) se refere a situações em que os operadores informam um horário incorreto de uma determinada ocorrência (ex. um desligamento). A adequação das condições de segurança (c) se refere à inclusão de um equipamento já impedido por outra SI em um outro procedimento, sem a ciência do estado operativo do mesmo. Já a inclusão de serviço dentro do prazo (d) se refere, por exemplo, à falta de aproveitamento de uma parada já autorizada para um mesmo equipamento e não verificada pelo operador no momento de criação de uma nova SI.

Para os casos (a) e (b), com os sistemas (aplicações) computacionais integrados, grande parte dos dados visualizados em um sistema poderiam ser preenchidos automaticamente em outro. Já para situações como as descritas em (c) e (d), além da integração dos sistemas através de uma plataforma de software como a que propomos aqui (um *middleware*), faz-se necessário que esta mesma plataforma possa inferir contexto a partir de dados provenientes dos diferentes sistemas (aplicações) e, então, apresentar informações melhor contextualizadas aos usuários (operadores) por meio de alguma das aplicações integradas. Desta forma, será possível incrementar o comportamento de qualquer aplicativo usado pelos operadores de tempo real, bem como pelos usuários das áreas de pré- e pós-operação, uma vez que as aplicações são informadas, de maneira automática, sobre o contexto dos eventos, como por exemplo, se um desligamento ou normalização de uma linha de transmissão é parcial ou total.

### III. PANORAMA TECNOLÓGICO

Nesta seção são discutidas tecnologias mais usadas no trabalho, que são a sensibilidade a contexto (do Inglês, *Context Awareness*), que permite a adaptação de sistemas computacionais a contextos (situações) estabelecidos dinamicamente, e *middleware*, que é uma plataforma de software para aumentar o grau de abstração na comunicação entre sistemas distribuídos e, possivelmente, heterogêneos.

#### A. Sensibilidade a Contexto

**Contexto** pode ser definido como um grupo de influências e condições que tragam relevância à aplicação, tornando uma situação única e compreensível [5]. Também pode ser considerado como qualquer informação que caracterize uma situação de uma entidade, podendo a entidade ser representada por uma pessoa, um lugar ou um objeto [2].

Estas entidades devem ser consideradas importantes para o usuário e para os sistemas que delas tratam. A utilização de contexto, no âmbito da computação, investiga o uso das informações presentes na interação entre pessoas e computadores, com o objetivo de melhorar a qualidade da comunicação entre o ser humano e sistemas computacionais. Tais informações, por vezes desconsideradas do processo de interação, são denominadas de informações contextuais, que contêm **elementos contextuais** (do Inglês, *Contextual Element*) e podem ser utilizadas como fontes de conhecimento pelos sistemas [5]. Quando um contexto passa a ser compreendido, um sistema pode, em diversas situações e condições, mudar sua sequência de ações, interações e o tipo de informação a ser fornecida aos usuários. Desta forma, o sistema passa a ser considerado **sensível a contexto**.

Diversas propostas são encontradas na literatura para a representação de informações contextuais (ex. grafos contextuais, par chave-valor, ontologias, entre outras) [6]. A técnica de modelagem par chave-valor utiliza uma estrutura de dados mais simples para representar o contexto, o qual é modelado por meio de pares compostos por uma chave, que identifica o atributo de contexto, e por um valor associado a essa chave. Grafos contextuais também podem ser utilizados para apoiar a modelagem do comportamento de um sistema sensível a contexto [5].

Através do uso de regras de inferência é possível determinar o comportamento e o funcionamento de uma aplicação, usando elementos contextuais, como por exemplo, períodos de manutenção programada, localização de uma instalação e estado operativo de um equipamento. A Fig. 1 ilustra a diferença entre aplicações tradicionais e aplicações sensíveis a contexto.

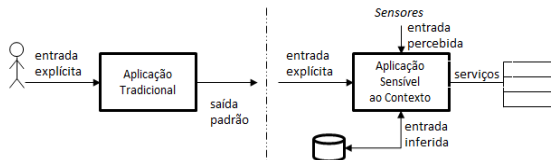


Figura 1. Diferenças entre os tipos de aplicações quanto a entradas e saídas. Adaptado de [5].

Através das informações contextuais (ex. percebidas, inferidas), a aplicação pode enriquecer semanticamente a solicitação explícita do usuário e com isso, fornecer e executar serviços mais próximos às suas necessidades.

### B. Middleware

**Middleware** é definido como uma camada de software que fornece uma abstração de programação, escondendo a heterogeneidade de redes, hardware, sistema operacional e linguagens de programação.

Integrar sistemas computacionais é fazer com que aplicações distintas trabalhem em conjunto através de suas funcionalidades para produzir um resultado em comum. O grande desafio está em integrar aplicações que foram desenvolvidas por empresas diferentes, tecnologias e épocas distintas [7].

Existem na literatura trabalhos, como o de C. de S. Penin, W. Sybine, C. M. Matayoshi e F. C. S. Cerdan [8], que propõem a utilização de um barramento de comunicação para a integração de aplicações utilizando o modelo CIM (*Common Information Model*), através do uso de conectores. Outros trabalhos [9], [10] utilizam um *Web Service RESTful*, intitulado *RESTful WS-GDA*, que encapsula o padrão *IEC 61970-403 Generic Data Access (GDA)*, demonstrando uma maneira de permitir a integração de dados de sistemas de gerenciamento de energia. A Fig. 2 apresenta *middleware*, ou barramento de serviços – ESB (*Enterprise Service Bus*), para integração de aplicações distintas do setor elétrico.

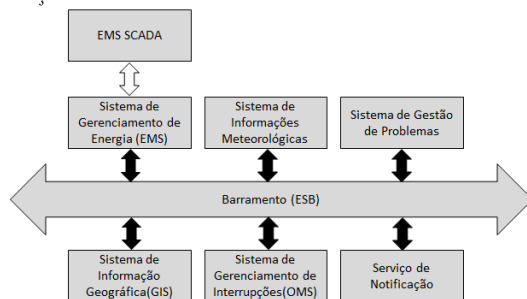


Figura 2. Middleware de integração de aplicações de um centro de controle. Adaptado de [10]

Utilizar um *middleware* permite a integração de sistemas com pouca intervenção na lógica do negócio das aplicações e com fraco acoplamento. Fraco acoplamento é um dos principais requisitos na integração de sistemas legados, como os muitos que são usados no

setor de energia elétrica. A Fig. 3 ilustra a diferença entre (a) **forte acoplamento**, onde os sistemas se comunicam sem intermediários, o que demanda maior intervenção nos seus códigos, e (b) **fraco acoplamento**, em que os sistemas se integram através de um barramento de comunicação (*middleware*).

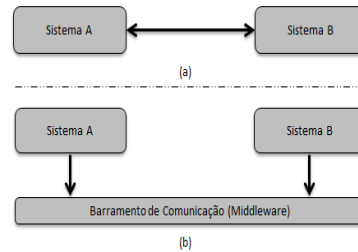


Figura 3. (a) Forte acoplamento (b) Fraco acoplamento

Através do uso de *Web Services* como interface padrão para os serviços do *middleware*, permite-se interoperabilidade, flexibilidade, escalabilidade para atender à integração de soluções distintas, independentemente do padrão adotado, seja ele *SOAP* ou *RESTful* [11]. Os padrões *Web Services* formam uma representação extensível e amplamente utilizada, fornecendo uma abordagem adequada de baixo acoplamento para a integração de aplicações de software das empresas do setor de energia elétrica [12].

### IV. UM MIDDLEWARE SENSÍVEL A CONTEXTO PARA INTEGRAÇÃO DE SISTEMAS ELÉTRICOS

Esta seção apresenta a proposta da arquitetura do *middleware* e seus principais componentes, organizados de forma a prover serviços relativos a contexto, além de outros serviços comuns.

#### A. Arquitetura

A Fig. 4 apresenta a arquitetura do *middleware* baseada em V. Raychoudhury, J. Cao, M. Kumar, D. Zhang [13], que propõe um modelo de referência de um *middleware* para computação pervasiva.

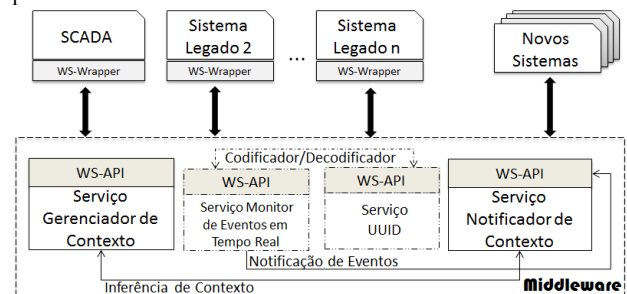


Figura 4. Arquitetura do Middleware para Integração Sensível a Contexto

Seguindo o modelo de referência citado [13], o *middleware* fornece serviços responsáveis pelo gerenciamento (aquisição, modelagem e raciocínio) e notificação de contexto (*Context Management Service* e *Context Notification Service*,

respectivamente), e serviços auxiliares (*Real Time Event Monitor* e *UUID Service*). Todos os serviços do *middleware* são especificados como Web Services (WS) no padrão SOAP, e a publicação de seus contratos via WS-API - API (*Application Programming Interfaces*) descritas em WSDL (*Web Services Description Language*), uma linguagem baseada em XML utilizada para descrever a funcionalidade de Web Services.

### B. Serviços

O serviço *Context Management Service* (Fig. 5) processa contexto, adotando o padrão *Domain-focused modelling* para a representação do conhecimento [14]. É utilizada uma execução de regras com suporte a eventos [15], através da engine *JBoss Drools*<sup>1</sup>, para criação de regras de produção.

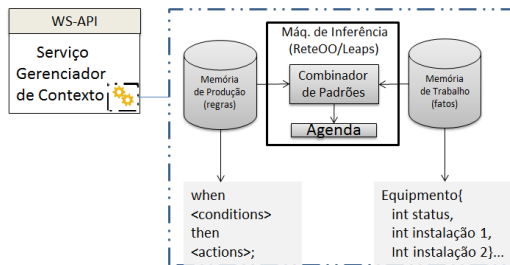


Figura 5. Arquitetura do Serviço de Gerenciamento de Contexto com Drools

O serviço *Context Notification Service* é responsável por gerenciar mensagens com as informações contextuais processadas pelo *Context Management Service*. Qualquer informação contextual é trocada entre as partes por meio de uma tecnologia tipo *publish/subscribe* (ex. *Java Message Service - JMS*).

O serviço *UUID Service* converte para um formato/identificador único (*UUID – Universally Unique Identifier*), segundo o modelo CIM (IEC 61970-301), os diferentes ID de um mesmo equipamento encontrados nos sistemas a serem integrados (ex. um disjuntor tem ID=xxx em um sistema A e ID=yyy em um sistema B). Outro serviço, o *Real Time Event Monitor*, monitora os eventos em tempo real captados por um sistema SCADA, a fim de identificar bloqueios, desligamentos e normalizações de um equipamento ou função de transmissão.

Como não havia o *middleware* antes dos sistemas de software existentes, a integração destes sistemas (legados) deve ser feita através de *wrappers*, que são pequenas peças de software desenvolvidas para serem acopladas a tais sistemas, permitindo que acessem as API do *middleware*. Novos sistemas (aplicações) não precisam de *wrappers*, pois podem ser

programados diretamente para as API já publicadas.

### V. PROTÓTIPO

Como prova de conceito, um protótipo (Fig. 6) foi desenvolvido para integrar as versões simuladas de um sistema de solicitação de impedimentos (*S\_LS1*) e de um sistema de registro de ocorrências (*S\_LS2*) através do *middleware* apresentado na seção anterior (IV). As versões usadas simulam o funcionamento de sistemas (legados) em produção na empresa CTEEP. Além disso, foram utilizados arquivos de log do sistema *SCADA* utilizado na CTEEP (*SAGE/SCADA – Sistema Aberto de Gerenciamento de Energia/Supervisão e controle de redes elétricas*<sup>2</sup>), com 644.627 eventos reais que aconteceram em 30 dias de 2013.

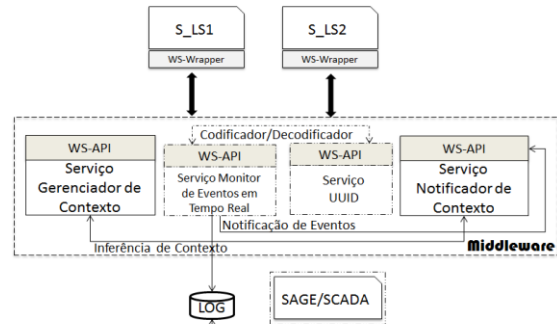


Figura 6. Arquitetura da prova de conceito.

Até o momento da escrita deste artigo, foi verificada a inferência de contextos referentes a dois cenários: 'identificação de desligamento' e 'liberação programada com desligamento', como descritos a seguir.

#### A. Cenário 1: Identificando desligamentos

Quando eventos de desligamento de linhas de transmissão ocorrem, o operador de tempo real se depara com uma gama de informações (na maioria das vezes, dados brutos), de sistemas distintos, para inferir qual o contexto dos eventos registrados, por exemplo, pelo sistema *SAGE/SCADA* (vide Ev1 e Ev2 na Fig. 7). Nesta simulação, o serviço *Real Time Event Monitor* identificou automaticamente todos os eventos de desligamento (4.194), eventos de normalização (8.405) e eventos de bloqueio (3.139) ocorridos, sem que o operador tivesse que visualizar o monitor de alarmes do *SAGE/SCADA*, por exemplo.

<sup>2</sup> <http://www.sage.cepel.br/>

<sup>1</sup> <http://www.jboss.org/drools/>

```

C:\Windows\system32\cmd.exe - java -jar monitor_tr-1.0.jar
00:54:18 BBAR_13T.GERALR25.TCMD BAR_138.GERAL_R25.COMANDO
Desligamento <ligado-desligado>
Numero da linha: 10
00:54:46 BBAB_13E.BAR1_DJ7.5200 BAB.138.BAR-1.DJ-7.
Desligado
Numero da linha: 11
00:54:48 CSJ2E13A.SAJ.MUA SJB2.138.SAJ.MUA
Violou advert. sup. <70.09>
Numero da linha: 12
00:54:51 SPBAB-1BAR-1 DESLIG
BARRA BONITA
Numero da linha: 13
00:54:58 BBAB_13E.BAR2_DJ21.EST BAB.138.BAR-2.DJ-21.
Desligado
Numero da linha: 14
00:54:59 SPBAB-1BAR-2 DESLIG
BARRA BONITA
Numero da linha: 15
00:55:02 CSTO_44A.RE1.TMPEA ST0.440.RE-1.Temp Enrolan FA
Violou advert. sup. <100.56>

```

Figura 7. Exemplo de log do SCADA registrando um desligamento

Neste cenário, um desligamento de um equipamento pode ocorrer de maneira forçada ou não forçada, também podendo ser parcial ou total. Com a utilização do serviço *Real Time Event Monitor* do *middleware*, identificou-se os eventos de desligamento parcial de um equipamento (Fig. 7):

- Ev1 - desligamento parcial lado BAR1
- Ev2 - desligamento parcial lado BAR2

Sem o uso do *middleware*, para inferir o contexto destes eventos independentes, o operador precisa ter conhecimento da infraestrutura de transmissão (instalações, BAY etc.) e/ou consultar o sistema de monitoramento (ex. um diagrama unifilar) para verificar se existe algum relacionamento entre os eventos.

Entretanto, com o uso do *Context Management Service*, que é acionado pelo *Real Time Event Monitor Service*, inferiu-se automaticamente o contexto dos desligamentos, ora tratados como parciais, ao executar a regra de contexto “Desligamento Total de uma LT” descrita em *Drools* (Fig. 8). O evento Ev2, na verdade, está relacionado ao evento Ev1, por se tratar de outro lado (BAR2) do mesmo equipamento (SPBAB-1). Desta forma, a regra de contexto concluiu ser um ‘desligamento total’ do equipamento (lados BAR1 e BAR2), embora os eventos tenham acontecido em momentos distintos, registrados no log pelos desligamentos parciais Ev1 e Ev2.

```

rule "Desligamento total de uma LT"
no-loop
when
//conditions
$desligamento : Desligamento( ativo.cod_instalacao1 != "" && ativo.cod_instalacao2 != "" )
then
//actions
ContextManager cm = ContextManager.getInstance();
Contexto ctx = new Contexto("É um desligamento total do equipamento
$desligamento.getEquipamento() dos lados: "+
$desligamento.getAtivo().getCod_instalacao1() + "-" +
$desligamento.getAtivo().getCod_instalacao2());
cm.adicionarContexto(ctx);
end

```

Figura 8. Regra para inferência de desligamento total utilizando Drools

Assim sendo, o operador ou sistema interessado, não precisou consultar outros sistemas para entender os eventos, e recebeu notificações através do *Context Notification Service* com o contexto dos eventos, como apresentado na Fig. 9.

MODELO - NOTIFICAÇÃO	MODELO - NOTIFICAÇÃO												
<table border="1"> <tr><td>Equipamento</td><td>_F541841f-c449-43bf-8533-dfaa19fc5a4a</td></tr> <tr><td>Descrição de Evento</td><td>DESILIGOU LADO BARRA BO...</td></tr> <tr><td>Data/Hora</td><td>2013-09-17 00:54:51</td></tr> </table>	Equipamento	_F541841f-c449-43bf-8533-dfaa19fc5a4a	Descrição de Evento	DESILIGOU LADO BARRA BO...	Data/Hora	2013-09-17 00:54:51	<table border="1"> <tr><td>Equipamento</td><td>_768cc5c9-7c9f-4254-9584-4c53e79b6353</td></tr> <tr><td>Descrição de Evento</td><td>DESILIGOU LADO BARRA BO...</td></tr> <tr><td>Data/Hora</td><td>2013-09-17 00:54:59</td></tr> </table>	Equipamento	_768cc5c9-7c9f-4254-9584-4c53e79b6353	Descrição de Evento	DESILIGOU LADO BARRA BO...	Data/Hora	2013-09-17 00:54:59
Equipamento	_F541841f-c449-43bf-8533-dfaa19fc5a4a												
Descrição de Evento	DESILIGOU LADO BARRA BO...												
Data/Hora	2013-09-17 00:54:51												
Equipamento	_768cc5c9-7c9f-4254-9584-4c53e79b6353												
Descrição de Evento	DESILIGOU LADO BARRA BO...												
Data/Hora	2013-09-17 00:54:59												

Figura 9. Notificação de desligamento (a) lado BAR1 (b) lado BAR2

Os dados são gravados em um arquivo XML (Fig. 10) com as informações relacionadas ao desligamento. Sendo assim, o contexto é publicado, permitindo que seus dados sejam aproveitados, por exemplo, no registro de um fato de inicialização ou de ocorrências no sistema *S\_SL2* (Fig. 6).

```

<Desligamento>
<ativo>
<id_ativo> 768cc5c9-7c9f-4254-9584-4c53e79b6353</id_ativo>
<cod_instalacao>BAR2</cod_instalacao>
<indicadorPS>S</indicadorPS>
</ativo>
<data>2013-09-17</data>
<hora>00:54:59</hora>
<descricao>DESILIGOU LADO BARRA BONITA</descricao>
</Desligamento>
<listaContexto>
<Contexto>
<observacao>É um desligamento total do equipamento SPBAB-1lados BAR1 e BAR2.</observacao>
</Contexto>
</listaContexto>

```

Figura 10. Exemplo de trecho do XML com o contexto do desligamento inferido

## B. Cenário 2: Liberação programada com desligamento

Neste cenário, em seguida ao desligamento ocorrido, o operador, através do uso do simulador do sistema de solicitação de impedimentos (*S\_SL1*), procedeu com uma liberação programada com desligamento, referente ao equipamento SPBAB-1. Para isto, *S\_SL1* executa uma chamada ao método *informPlannedRelease*, disponibilizado pelo serviço *Context Management Service*.

Antes da publicação das informações de contexto desta liberação para os assinantes interessados, o serviço *Context Notification Service* solicita ao *Context Management Service* que infira o contexto dos eventos capturados pelo SAGE/SCADA (no protótipo os eventos são gravados em um log lido pelo *Real Time Event Monitor*). O *Context Management Service*, então, infere que o desligamento não é forçado, de acordo com as regras descritas em *Drools* (ex. ‘desligamento’ E ‘liberação programada com desligamento’ IMPLICA ‘desligamento não forçado’).

Sem o *middleware* integrando os sistemas e processando contextos, o operador precisaria consultar os mesmos sistemas (*S\_LS1*,

SAGE/SCADA) para reunir dados, entendê-los, formando o contexto do momento, e cadastrar manualmente os dados relativos ao fato de finalização no sistema simulado (S\_LS2), o que poderia incorrer nos erros já mencionados na seção I. Com o *middleware*, o cadastro é feito automaticamente no sistema de registro de ocorrências (S\_LS2, neste caso).

## VI. CONCLUSÕES

Até o presente momento, os resultados preliminares desta pesquisa suportam a ideia de que é possível a utilização de um *middleware* com serviços sensíveis ao contexto, para integrar os diferentes sistemas de gerenciamento da transmissão de energia de um moderno centro de controle, minimizando o tempo gasto para o cadastro de ocorrências e evitando assim o cancelamento de SI.

A arquitetura adotada permite uma adaptação mínima em sistemas legados, devido ao acoplamento de *wrappers* para acessar os serviços do *middleware*. Os serviços de integração sensível ao contexto permitiram a inferência de contextos relativos a desligamentos, normalizações e liberações programadas de equipamentos, a fim de notificar aplicações e operadores interessados através do modelo de modelo *publish/subscribe*.

Os resultados até aqui demonstram que o monitoramento “visual/manual” e contínuo de alarmes e várias telas de diferentes sistemas podem ser reduzidos, permitindo aos operadores reduzir os cancelamentos de SI pelos motivos já relacionados na seção II.

Em resumo, o *middleware* tem três características principais: (i) permite a integração de aplicações de gerenciamento de transmissão de energia, (ii) utiliza a sensibilidade ao contexto para, automaticamente, inferir e notificar os contextos relacionados ao estado operativo dos equipamentos; (iii) utiliza um mecanismo de regras de produção que permite a adição e inferência de novos contextos em tempo de execução. Desta forma, pode-se afirmar que o modelo proposto neste trabalho, permite integrar os sistemas utilizados nos modernos centros de controle, podendo diminuir a carga de trabalho dos operadores e incrementando a segurança no planejamento e execução das atividades diárias.

Trabalhos futuros incluem a melhoria da funcionalidade de ambos os serviços de contexto e aplicações, e mais testes para medir a quantidade de ocorrências não forçadas, ocorrências forçadas e bloqueios identificados

corretamente para cadastro automático de ocorrências.

## REFERÊNCIAS

- [1] R. D. Masiello, “Computers in power: a welcome invader,” *IEEE Spectrum*, vol. 22(2), pp. 51–59, Feb. 1985.
- [2] A. K. Dey, D. Salber and G. D. Abowd, “A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware application,” *HCI*, vol. 16(2-4), pp. 91–166, 2001.
- [3] Y. Yan, Y. Qian, H. Sharif and D. Tipper, “A Survey on Smart Grid Communication Infrastructures: Motivations, Requirements and Challenges,” *IEEE Comm. Surveys & Tutorials*, vol. 15(1), pp. 5–20, 2013.
- [4] P. Zhang, F. Li and N. Bhatt, “Next-generation monitoring, analysis, and control for the future smart control center,” *IEEE Transactions on Smart Grid*, vol. 1(2), pp. 186–192, Sept. 2010.
- [5] V. Vieira, P. Tedesco e A. C. Salgado, “Modelos e processo para o desenvolvimento de sistemas sensíveis ao contexto,” in *Jornadas de Atualização em Informática (JAI)*, vol. 1, A. P. de L. F. de Carvalho, T. Kowaltowski (Org.), Ed. Porto Alegre: SBC, 2009, pp. 381-431.
- [6] V. Vieira, D. Souza, A. C. Salgado e P. Tedesco, “Uso e Representação de Contexto em Sistemas Computacionais,” in *Tópicos em Sistemas Interativos e Colaborativos*, C. A. C. Teixeira, C. R. G. de Farias, J. C. Leite e R. O. Prates (Org.), Ed. São Carlos: UFSCAR, 2006, pp.127-166.
- [7] G. Hohpe and B. Woolf, *Enterprise integration patterns: Designing, building, and deploying messaging solutions*. Boston: Addison-Wesley Longman Publishing Co., Inc., 2004, pp. 63-74.
- [8] C. de S. Penin, W. Sybine, C. M. Matayoshi and F. C. S. Cerdan, “Common Information Model: A Bus Service for Electric Calculations in AES Eletropaulo,” *JEPE*, vol. 6, pp. 965–971, 2012.
- [9] L. Imre, E. Varga, A. Erdeljan and M. Gavrić, “RESTful web services and the Common Information Model (CIM),” in *Proc. IEEE International Energy Conference and Exhibition, EnergyCon 2010*, pp. 716-721.
- [10] E. Varga, I. Lendak, M. Gavrić and A. Erdeljan, “Applicability of RESTful Web Services in Control Center Software Integrations,” in *Proc. 2011 IEEE International Conference on Innovations in Information Technology*, pp. 282–286.
- [11] F. Belqasmi, J. Singh, S. Y. B. Melhem and R. H. Glitho, “Soap-based vs. restful web services: A case study for multimedia conferencing,” *IEEE Internet Computing*, vol. 16(4), pp. 54–63, Jul. 2012.
- [12] Q. Chen, H. Ghenniwa and W. Shen, “Web-services infrastructure for information integration in power systems,” in *2006 IEEE Power Engineering Society General Meeting*, pp. 1-8.
- [13] V. Raychoudhury, J. Cao, M. Kumar and D. Zhang, “Middleware for pervasive computing: A survey,” *Pervasive and Mobile Computing*, vol. 9(2), pp. 177–200, Apr. 2013.
- [14] C. Bettini, O. Brdiczka, K. Henriksen, J. Indulska, D. Nicklas, A. Ranga-Nathan and D. Riboni, “A survey of context modeling and reasoning techniques,” *Pervasive and Mobile Computing*, vol. 6(2), pp. 161–180, Apr. 2010.
- [15] A. Badii, M. Crouch and C. Lallah, “A context-awareness framework for intelligent networked embedded systems,” in *Proc. 2010 3rd CENTRIC’10*, IEEE Computer Society, pp. 105–110, Aug. 2010.