



**XXIII SNPTEE  
SEMINÁRIO NACIONAL  
DE PRODUÇÃO E  
TRANSMISSÃO DE  
ENERGIA ELÉTRICA**

FI/GTL/32  
18 a 21 de Outubro de 2015  
Foz do Iguaçu - PR

**GRUPO – XV**

**GRUPO DE ESTUDO DE SISTEMAS DE INFORMAÇÃO E TELECOMUNICAÇÃO PARA SISTEMAS ELÉTRICOS - GTL**

**INTEGRANDO SISTEMAS DE INFORMAÇÃO NO SETOR ELÉTRICO ATRAVÉS DE SENSIBILIDADE A CONTEXTO**

**Virgínia A. C. Sgotti (\*)**  
**In Forma Software Ltda**

**Jonysberg P. Quintino**  
**Centro de Informática/ UFPE**

**Carlos A. G. Ferraz**  
**Centro de Informática/ UFPE**

**Marcos Bertinotti**  
**CTEEP**

**RESUMO**

O monitoramento de desempenho da rede elétrica frequentemente envolve a integração de vários sistemas e é tarefa rotineira dos operadores dos centros de controle. Entretanto, em geral, 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. Este trabalho apresenta uma plataforma de software capaz de inferir contexto e integrar sistemas. Um protótipo funcional foi implementado e os resultados preliminares apontam para uma maior eficiência na operação de um centro de controle.

**PALAVRAS-CHAVE**

Integração de sistemas, *Middleware*, Sensibilidade a contexto

**1.0 - 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. 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, exigindo a visualização de informações em diferentes telas de sistemas (aplicações) de software distintos, como demonstrado na Figura 1, para o entendimento do contexto sobre o estado operativo dos equipamentos, e a redigitação de dados em comum entre os sistemas. Tipicamente esta integração “manual-visual” tem por consequência inconsistências geradas por erros de digitação ou de interpretação de informações, às quais podem levar a tomada de decisão inadequada. 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.

(\*) Rua Domingos José Martins, n° 75 – sala 302 – CEP 50.030-200 Recife, PE, – Brasil  
Tel: (+55 81) 2137-6767 – Fax: (+55 81) 2137-6777 – Email: vsgotti@informa.com.br



**Figura 1** - Exemplo de um Centro de Supervisão e Controle

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. Por exemplo, um desligamento detectado pelo sistema SCADA, mas que não tem qualquer Solicitação de Intervenção planejada para o horário, faz com que o middleware infira o contexto de que o desligamento não foi programado. Com este contexto, o middleware consulta o sistema de descargas atmosféricas para verificar se o desligamento pode ter sido causado por um raio. Desta forma, contextos vão sendo montados e disponibilizados para usuários dos sistemas integrados, de forma a facilitar a tomada de decisão e registro dos eventos. Da mesma forma, através de contextos, o middleware consegue agrupar os eventos relacionados, como desligamentos de equipamentos primários e secundários, normalizações, liberações, dentre outros, que têm a ver com a mesma ocorrência..

Em Y. Yan, Y. Qian, H. Sharif and D. Tipper [12] é apresentada uma visão, originalmente proposta por P. Zhang, F. Li e N. Bhatt [13], de centros de controle do futuro, também chamados de *smart control centers*, onde uma de suas 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 processar informação rapidamente.

Neste artigo, adicionalmente propomos o uso de sensibilidade a contexto ou computação sensível a contexto (definida em 2.2) 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 referentes a estas 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 permitiu 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 2.0 - apresenta o panorama tecnológico envolvido nesta proposta; a seção 3.0 - explica a arquitetura proposta para o middleware sensível ao contexto; a seção 4.0 apresenta o estudo de caso utilizado para a prova de conceitos e, por fim, as conclusões são apresentadas na seção 5.0 - .

## 2.0 - FUNDAMENTAÇÃO TEÓRICA

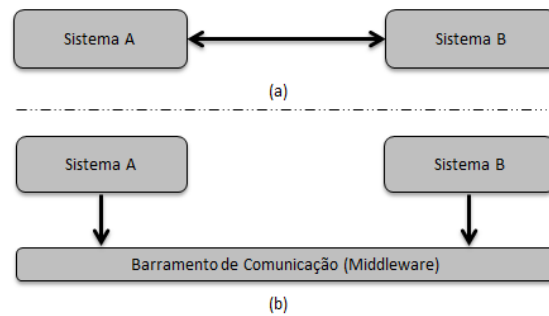
Esta sessão aborda o referencial teórico encontrado na literatura com os principais conceitos relacionados à aplicação de computação nos Sistemas Elétricos, Sensibilidade a Contexto e Middleware, como também, trabalhos já propostos que utilizam alguns desses conceitos..

### 2.1 Integração de Sistemas

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 [2].

Existem diversas abordagens tecnológicas na literatura e no mercado, para possibilitar uma integração de sistemas computacionais, entre elas, o uso de um *middleware*. Para Coulouris et al.[3], *middleware* se aplica a uma camada de software que disponibiliza serviços e fornece uma abstração de programação, escondendo a heterogeneidade das redes, hardware, sistema operacional e linguagens de programação.

Ao utilizar um *middleware*, permite-se a integração de sistemas com pouca intervenção na lógica de negócio das aplicações, através do conceito de fraco acoplamento. Fraco acoplamento é um dos principais requisitos na integração de sistemas legados, como os muitos que são usados em diversos setores, tal como no setor de energia elétrica. A Figura 2 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 (ex. *middleware*).



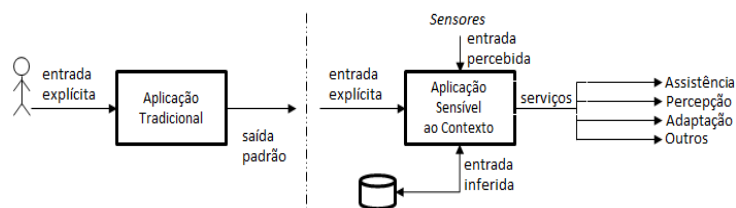
**Figura 2 - (a) Forte acoplamento (b) Fraco acoplamento**

Unificar soluções heterogêneas utilizando *middleware* e sensibilidade a contexto constituem um grande desafio, uma vez que existe uma grande diversidade de tecnologias e aplicabilidade para estes conceitos. Como exemplo, a computação ubíqua e pervasiva é uma das áreas que mais utiliza tais funcionalidades, com o objetivo principal criar um ambiente inteligente com dispositivos embutidos de computação em rede, proporcionando aos usuários um contínuo acesso aos serviços prestados [4].

## 2.2 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 [6]. 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. A Figura 3 ilustra a diferença entre aplicações tradicionais e aplicações sensíveis a contexto.



**Figura 3 - Diferenças entre os tipos de aplicações quanto a entradas e saídas. Adaptado de [5].**

Diversas propostas são encontradas na literatura para a representação de informações contextuais [7] (ex. grafos contextuais, par chave-valor, ontologias). 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[11].

Em Raychoudhury et al. [8] é apresentado um modelo de referência com as principais características básicas que devem ser implementadas para o desenvolvimento de um *middleware* sensível a contexto. Segundo este trabalho, a sensibilidade a contexto pode ser classificada em quatro áreas primárias: (i) aquisição de contexto, é um pré-requisito para aplicações sensíveis ao contexto e refere-se a um processo de obtenção de contextos a partir de

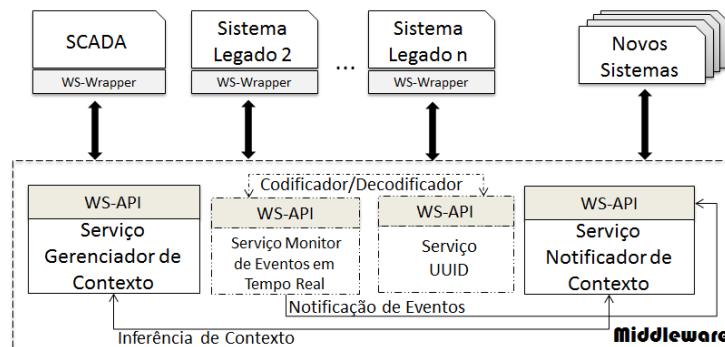
várias fontes; (ii) armazenamento de contexto, para gerenciar grandes volumes de dados e seus relacionamentos; (iii) modelagem de contexto, para utilizar diversas abordagens para a representação contextual das informações; (iv) inferência de contexto, processamento para obtenção de informações de alto nível de contexto implícito, baseado em informações de contexto explícita de baixo nível.

### 3.0 - UM MIDDLEWARE SENSÍVEL A CONTEXTO PARA INTEGRAÇÃO DE SISTEMAS DE SUPERVISÃO E CONTROLE

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

#### 3.1 Arquitetura

A Figura 4 apresenta a arquitetura do *middleware* baseada em Raychoudhury et al. [8], que propõe um modelo de referência de um *middleware* para computação pervasiva.

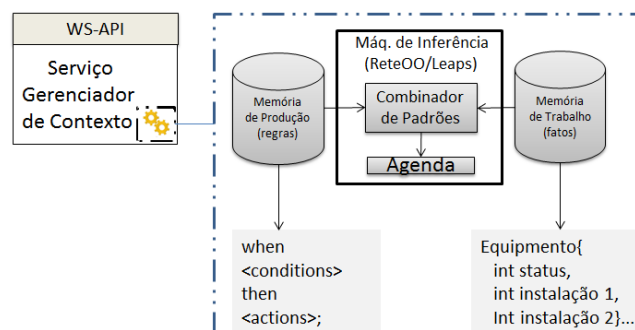


**Figura 4 - Arquitetura do Middleware para Integração Sensível a Contexto**

O *middleware* proposto fornece serviços responsáveis pelo gerenciamento (aquisição, modelagem e raciocínio) e notificação de contexto (*Serviço Gerenciador de Contexto* e *Serviço Notificador de Contexto*, respectivamente), e serviços auxiliares (*Serviço Monitor de Eventos em Tempo Real* e *Serviço UUID*). 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.

#### 3.2 Serviços

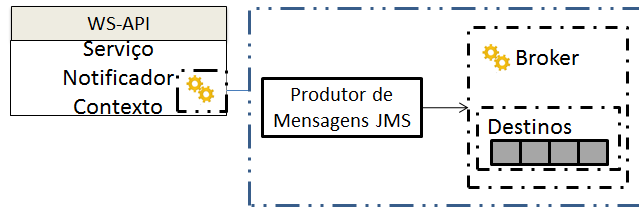
O serviço Gerenciador de Contexto (Figura 5) processa contexto, adotando o padrão *Domain-focused modelling* para a representação do conhecimento [9]. É utilizada uma execução de regras com suporte a eventos [10], através da engine *JBoss Drools*<sup>1</sup>, para criação de regras de produção.



**Figura 5 - Arquitetura do Serviço de Gerenciamento de Contexto com Drools**

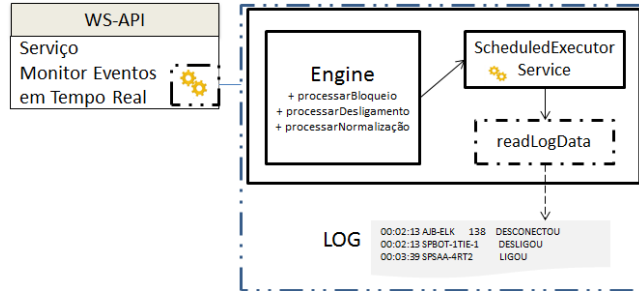
O serviço Notificador de Contexto (Figura 6) é responsável por gerenciar mensagens com as informações contextuais processadas pelo Serviço Gerenciador de Contexto. Qualquer informação contextual é trocada entre as partes por meio de uma tecnologia tipo *publish/subscribe* (ex. *Java Message Service - JMS*).

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



**Figura 6** - Arquitetura do Serviço Notificador de Contexto

O serviço *UUID* 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 monitor de eventos em Tempo Real, que 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.

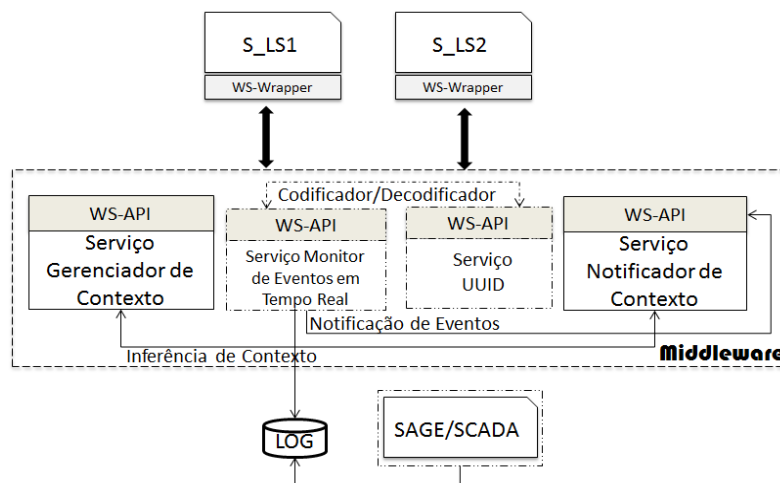


**Figura 7** - Arquitetura do Serviço Monitor de Eventos em Tempo Real

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.

### 3.3 Implementação

Como prova de conceito, um protótipo (Figura 8) 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 (3.2). As versões usadas simulam o funcionamento de sistemas (legados) em produção na empresa CTEEP (Companhia de Transmissão de Energia Elétrica Paulista). 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.



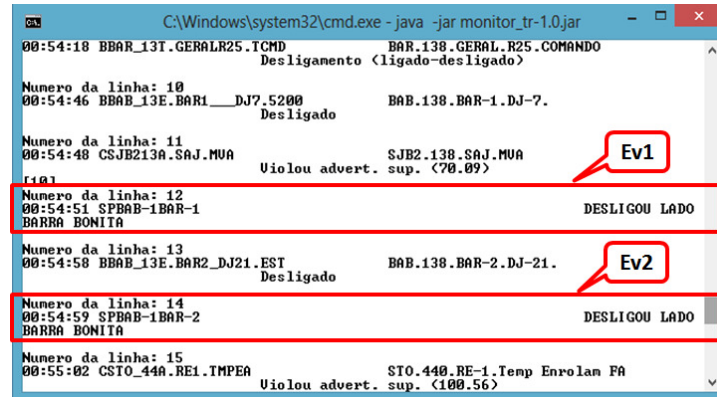
**Figura 8** - Arquitetura da prova de conceito

Para exemplificar o uso dos serviços do *middleware*, dentre diversos cenários e contextos inferidos em [11], demonstra-se a inferência de contextos referentes ao cenário de identificação de desligamento, como descrito a seguir.

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

### 3.4 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 Figura 9). 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.



**Figura 9** - 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 Monitor de Eventos em Tempo Real do *middleware*, identificou-se os eventos de desligamento parcial de um equipamento (Figura 9).(i) Ev1 - desligamento parcial lado BAR1;(ii)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 Serviço Gerenciador de Contexto, que é acionado pelo Serviço Monitor de Eventos em Tempo Real, 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*<sup>3</sup> (Figura 10). 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 10** - 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 Figura 11.

<sup>3</sup> [www.drools.org](http://www.drools.org)



MODELO - NOTIFICAÇÃO		MODELO - NOTIFICAÇÃO	
Equipamento	_f541641f-c449-43bf-8533-dfaa19fc5a4a	Equipamento	_768cc5c9-7c9f-4254-9584-4c53e79b6353
Descrição de Evento	Data/Hora	Descrição de Evento	Data/Hora
DESLIGOU LADO BARRA BO...	2013-09-17 00:54:51	DESLIGOU LADO BARRA BO...	2013-09-17 00:54:59

(a)

(b)

**Figura 11** - Notificação de desligamento (a) lado BAR1 (b) lado BAR2

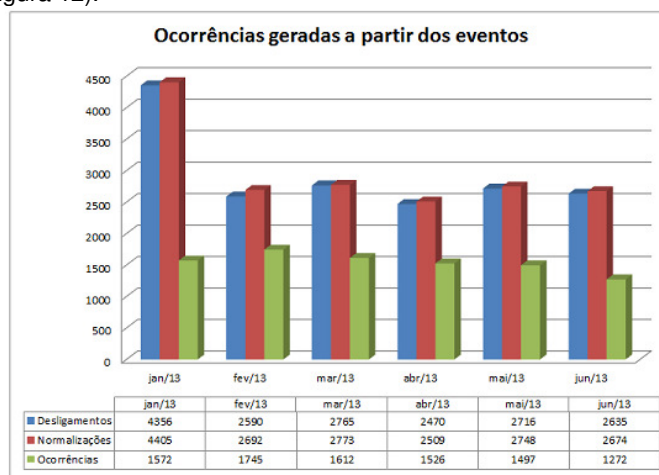
Os dados são gravados em um arquivo XML com as informações relacionadas ao desligamento. Sendo assim, o contexto é publicado, permitindo que seus dados sejam aproveitados, por exemplo, no registro de um fato de inicialização ou de ocorrências no sistema S\_SL2 (Figura 8). Na seção 0a seguir, será apresentado o estudo de caso aplicado na CTEEP e os resultados obtidos através do uso de dados reais dos sistemas utilizados na empresa.

#### 4.0 Estudo de Caso

Neste seção, é apresentado um resumo da análise de resultados, referentes aos testes executados no protótipo apresentado anteriormente, contemplando cinco cenários reais apresentados em [11]. Utilizou-se arquivos de LOG registrados pelo SCADA, em produção na CTEEP, no primeiro semestre do ano de 2013, como fonte de dados reais de um funcionamento de um moderno centro de operações, com uma média de 700.000 eventos registrados por mês.

#### 4.1 Resultados e Análises

Para permitir as análises a respeito da acurácia dos serviços que o middleware oferece, monitorou-se todos os arquivos de LOG, cedidos pela CTEEP, que foram gerados pelo SCADA durante o primeiro semestre do ano de 2013. O middleware identificou e notificou, uma média mensal de 3.000 eventos de desligamentos e normalizações. Assim como, uma média de 2.700 bloqueios mensais de equipamentos foram identificados. Este volume de eventos por fim, geraram automaticamente, uma média 1.500 ocorrências por mês no sistema gerenciador de ocorrências (Figura 12).



**Figura 12** - Ocorrências geradas automaticamente pelo middleware

Os dados oriundos da integração, análise e inferência de contextos, possibilitaram o preenchimento de 67% dos campos obrigatórios no sistema responsável pelo registro de ocorrências forçadas e 90% dos campos das ocorrências não forçadas.

A taxa de acerto do middleware, na amostra utilizada (78 liberações com desligamento em Jan/2013), para a inferência dos contextos de ocorrências não forçadas, em comparação com as mesmas inferências e registros manuais destas ocorrências, exercidas pelos usuários, foi de 100%, indicando uma boa assertividade dos contextos inferidos pelo middleware. O resultado apresentado sugere um forte indício de que os serviços do middleware estão corretos na inferência dos contextos avaliados. Portanto, as ocorrências geradas e pré-cadastradas, decorrentes destes eventos e contextos, poderão ser utilizadas de forma confiável durante o registro das ocorrências, quando do uso do middleware em um ambiente real.

#### 5.0 - CONCLUSÃO

O planejamento e execução de manutenções, como também, o monitoramento do estado operativo dos equipamentos são fatores importantes para a continuidade da operação e do contínuo fornecimento dos serviços prestados à população. Este trabalho apresentou um modelo de arquitetura de middleware para integração sensível ao contexto de sistemas de gerenciamento de energia elétrica, provendo para os usuários informações

relevantes, considerando que relevância depende das tarefas que o usuário está executando.

Através de dados e informações contextuais inferidas, as aplicações integradas pelo middleware sensível a contexto, puderam notificar os interessados em eventos diários como bloqueios, desligamentos, normalizações, liberações e normalizações programadas. Esta integração permitiu aos usuários a correta inferência do contexto dos eventos e o aumento da eficácia de suas atividades, como também, incrementar a segurança dos procedimentos de planejamento e execução das manutenções operativas através do uso da sensibilidade a contexto. Foi possível simular a rotina operacional de um centro de controle para testar o protótipo.

Comparando-se os resultados dos testes do middleware, com o que foi feito manualmente pelos operadores, foi alcançado um percentual de acerto significativo (100%) na inferência dos contextos para as ocorrências forçadas e não forçadas. Também se constatou que aproximadamente 67% dos campos das ocorrências forçadas e 90% das não forçadas foram preenchidos, automaticamente, pelos dados oriundos da integração e da inferência dos contextos.

Desta maneira, existe um forte indício de que o processo de registro de ocorrências, pelos operadores, poderá ser acelerado, uma vez que a maioria dos campos estará preenchida e os contextos já inferidos, permitindo minimizar os cancelamentos por erro nas informações, como também, eliminar o trabalho de redigitação de dados entre os sistemas envolvidos. Com base nestes resultados, pode-se afirmar que o modelo de arquitetura proposto neste trabalho permite integrar os sistemas utilizados nos modernos centros de controle das empresas de transmissão de energia, aumentando a eficácia e incrementando a segurança dos procedimentos de planejamento e execução das manutenções operativas através do uso da sensibilidade a contexto. Também poderá ser aplicado, com as devidas adaptações, a outros sistemas de gerenciamento como água, geração de energia ou outros setores.

## 6.0 - REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Masiello, R. D. (1985). Computers in power: A welcome invader: Supervisory computer systems cut utility costs by scheduling generating units, planning against outages, and coordinating the sale of power. *IEEE Spectrum*, 22(2), 51–59.
- [2] 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.
- [3] Coulouris, G., Dollimore, J., Kindberg, T., and Blair, G. (2011). *Distributed Systems: Concepts and Design*, volume 5 of *International computer science series*. Addison-Wesley.
- [4] Raychoudhury, V., Caob, J., Kumarc, M., and Zhangd, D. (2013). *Middleware for pervasive computing: A survey*. *Pervasive and Mobile Computing*, 9(2), 177–200.
- [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] 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.
- [7] 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.
- [8] Raychoudhury, V., Caob, J., Kumarc, M., and Zhangd, D. (2013). *Middleware for pervasive computing: A survey*. *Pervasive and Mobile Computing*, 9(2), 177–200.
- [9] 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.
- [10] 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.
- [11] Quintino, Jonysberg Peixoto. *Middleware sensível a contexto para integração de sistemas de gerenciamento de energia elétrica*. Recife, 2014, 115 f.: tab. Dissertação(Mestrado) - Universidade Federal de Pernambuco. CIN. Ciência da Computação, 2014.
- [12] 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.
- [13] 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.