

Brasília, 31 de agosto de 2007.

RELATÓRIO DE ATIVIDADES DE PROJETO DE PESQUISA Agosto/2007

Edgar Jhonny Amaya Simeón

FINATEC
ELETRONORTE
Universidade de Brasília – UnB

1. Objetivo

Apresentar o trabalho feito no quarto trimestre do projeto de pesquisa que é parte integrante do projeto de modernização da área de automação de processos da usina hidrelétrica de Balbina e Samuel, desenvolvido pelo programa de pesquisa e desenvolvimento tecnológico da empresa de geração e transmissão de energia elétrica, ELETRONORTE, em parceria com a FINATEC e a UnB.

2. Atividades realizadas

Podem ser listadas as seguintes atividades específicas:

- ✓ Papers submetidos e aceitos no COBEM 2007 - 19th International Congress of Mechanical Engineering, "Different Control Strategies used in Didactic Plant PD-3 Of Smar Through OPC Technology" e "An Intelligent Kernel for the Maintenance System of a Hydroelectric Power Plant".
- ✓ Paper Submetido e aceito no CIBIM8 – 8º Congresso Iberoamericano de Engenharia Mecânica – Cusco - Peru, "Sistema Inteligente de Manutenção Baseada em Condição para Usina Hidrelétrica de Balbina".
- ✓ Integração dos dados obtido através do servidor OPC com o Ikernel V3.0 em desenvolvimento, onde os dados de configuração host, progid e as tags são lidas do arquivo config.ini, os valores das tags obtidas são processadas por a maquina de inferência de JESS, e no futuro por o Fuzzy-JESS, Redes Neurais.
- ✓ Integração dos dados obtido através do servidor DB(Banco de Dados) com o Ikernel V3.0 em desenvolvimento, onde os dados de configuração e as tags são lidas do arquivo config.ini, os valores das tags obtidas são processadas por a maquina de inferência de JESS, e no futuro por o Fuzzy-JESS, Redes Neurais.

3. Integração do sistema IKernel com o Servidor OPC.

Para a Integração da comunicação através do servidor OPC com o Ikernel, grava-se no arquivo "config.ini" toda a configuração, host(nome ou IP do

computador onde encontra-se o O servidor PC), progid(o nome do servidor OPC), domain(Domínio onde encontra-se o computador onde esta o servidor OPC), user(O nome do usuário do computador), password(a senha do usuário do computador) e as tags(nomes das grandezas reconhecidas pelo servidor OPC).

O classe “configuration” do I-kernel lê toda a configuração do arquivo “config.ini” as tags lidas serão processadas pela classe “OPCProxy” do Ikernel e enviadas ao servidor OPC onde o servidor OPC retornara os valores das grandezas. Os valores das tags obtidas serão usados para o processamento inteligente da maquina de inferência das regras de produção usando o JESS, e futuramente por o Fuzzy-JESS ou Redes Neurais.

3.1 Arquivo de Tags

O arquivo “config.ini” tem a seguinte estrutura:

- Lista dos servidores OPC

```
[OPCServers]
dfi
df65
```

- Parâmetros de cada OPC server

```
[dfi]
progid = Smar.DfiOleServer.0
host = 164.41.17.129
domain = graco.unb.br
user = alvares
password = eletronorte2003
tags = dfiTags
```

```
[df65]
progid = Smar.DF65Server.1
host = 164.41.17.129
domain = graco.unb.br
user = alvares
password = eletronorte2003
tags = df65Tags
```

- Lista das tags que encontra-se em cada servidor OPC.

```
[dfiTags]
g1.t.metal.mguia.inf1 = FIT-31_AI1.PV.VALUE
g1.t.metal.mguia.inf2 = LIT-31_AI1.PV.VALUE
g1.t.metal.mguia.inf3 = TIT-31_AI1.PV.VALUE
g1.t.oleo.mguia.inf1 = FIT-32_AI1.PV.VALUE
g1.t.oleo.mguia.inf2 = TIT-32_AI1.PV.VALUE
g1.p.oleo.cuba = TIT-32_AI1.OUT.VALUE
g1.st.t.metal.mguia.inf1 = FIT-31_AI1.PV.STATUS
g1.st.t.metal.mguia.inf2 = LIT-31_AI1.PV.STATUS
g1.st.t.metal.mguia.inf3 = TIT-31_AI1.PV.STATUS
g1.st.t.oleo.mguia.inf1 = FIT-32_AI1.PV.STATUS
g1.st.t.oleo.mguia.inf2 = TIT-32_AI1.PV.STATUS
g1.st.p.oleo.cuba = TIT-32_AI1.OUT.STATUS
```

```
[df65Tags]
NivelBaixo = PD3.A.001.00
TempHiTqAquec = PD3.A.001.01
Emergencia = PD3.A.001.04
TempHiTqMist = PD3.A.002.05
LocalBBA1 = PD3.E.010.00
LocalBBA2 = PD3.E.010.01
Termostato = PD3.E.011.00
ChavedeNivel = PD3.E.011.01
SeloBBA1 = PD3.E.011.04
DefeitoBBA1 = PD3.E.011.05
SeloBBA2 = PD3.E.011.10
DefeitoBBA2 = PD3.E.011.11
AquecInhibido = PD3.S.012.00
BBA1Deslig = PD3.S.012.01
BBA2Deslig = PD3.S.012.02
BBA1 = PD3.S.012.03
BBA2 = PD3.S.012.04
Temp = PD3.S.012.05
Nivel = PD3.S.012.06
Sirene = PD3.S.012.10
```

3.2 Tela para escolher uma tag.

Na figura 1 mostra-se a tela de acordo a arvore hierárquica dos ativos da usina de Balbina.

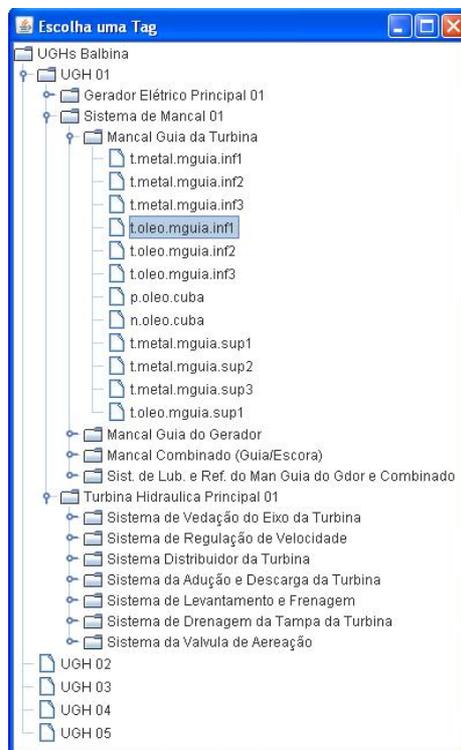


Fig. 1 – Tela para escolha de tags.

3.3 Tela de apresentação Inspeção de Variáveis.

A figura 2 mostra os valores obtidos do servidor OPC



Tag	Valor
clock	18:37:00
t.metal.mguia.inf1	128
t.metal.mguia.inf2	138.76472
t.metal.mguia.inf3	-14.89062
t.oleo.mguia.inf1	24,15805
t.oleo.mguia.inf2	4.0
t.oleo.mguia.inf3	1.0
p.oleo.cuba	325538.5
n.oleo.cuba	0.0
t.metal.mguia.sup1	24.603748
t.metal.mguia.sup2	1.0
t.metal.mguia.sup3	385612.28
t.oleo.mguia.sup1	24473.16

Fig. 2 - Tela dos valores das variáveis

4. Integração do sistema IKernel com o Servidor DB.

Esta aplicação desenvolvida em Java consiste em ler as tags gravadas no arquivo "config.ini", processar estas tags no programa em Java, enviar essas ao servidor DB e esperar o retorno dos valores para logo usar os valores obtidos no análise de tendência.

4.1 Arquivo de Tags

- Lista dos Servidores de Banco de Dados

```
[DBServers]
Assetview
```

- Parametros de cada Servidor DB

```
[Assetview]
user = sa
password = smar
DriveAddress = com.microsoft.sqlserver.jdbc.SQLServerDriver
ComAddress= jdbc:sqlserver://164.41.17.129:1092;databaseName=AssetView
tags = DBTags
```

- Lista das tags, pode ser real(do banco de dado especificado) ou simulado(valores armazenados no arquivo "values.ini").

```
[DBTags]
temp3 = real
temp4 = real
temp5 = real
temp6 = real
temp7 = real
temp8 = real
temp9 = real
temp10 = real
temp11 = real
temp12 = real
temp13 = real
temp14 = real
temp15 = real
temp16 = real
temp17 = real
temp18 = real
temp19 = real
temp20 = real
```

- Lista SELECT das tags a serem lidas do banco de dados na tabela e na posição.

```
[SELECT]
temp3 = SELECT * FROM ParametersHistory WHERE
CodParametersHistory='13'
temp4 = SELECT * FROM ParametersHistory WHERE
CodParametersHistory='14'
temp5 = SELECT * FROM ParametersHistory WHERE
CodParametersHistory='15'
temp6 = SELECT * FROM ParametersHistory WHERE
CodParametersHistory='16'
temp7 = SELECT * FROM ParametersHistory WHERE
CodParametersHistory='17'
temp8 = SELECT * FROM ParametersHistory WHERE
CodParametersHistory='18'
temp9 = SELECT * FROM ParametersHistory WHERE
CodParametersHistory='19'
temp10 = SELECT * FROM ParametersHistory WHERE
CodParametersHistory='20'
temp11 = SELECT * FROM ParametersHistory WHERE
CodParametersHistory='21'
temp12 = SELECT * FROM ParametersHistory WHERE
CodParametersHistory='22'
temp13 = SELECT * FROM ParametersHistory WHERE
CodParametersHistory='23'
temp14 = SELECT * FROM ParametersHistory WHERE
CodParametersHistory='24'
temp15 = SELECT * FROM ParametersHistory WHERE
CodParametersHistory='25'
temp16 = SELECT * FROM ParametersHistory WHERE
CodParametersHistory='26'
temp17 = SELECT * FROM ParametersHistory WHERE
CodParametersHistory='27'
temp18 = SELECT * FROM ParametersHistory WHERE
CodParametersHistory='28'
temp19 = SELECT * FROM ParametersHistory WHERE
CodParametersHistory='29'
temp20 = SELECT * FROM ParametersHistory WHERE
CodParametersHistory='30'
```

```
[INSERT]
DBinsert = INSERT INTO DBTagsInformation VALUES ('>?', '>?')
```

```
[UPDATE]
temp4 = UPDATE DBTagInformation SET Value = '>?' WHERE Indice = '2'
```


Anexo 1

Papers Submetido ao COBEM2007 e CIBIM8

AN INTELLIGENT KERNEL FOR THE MAINTENANCE SYSTEM OF A HYDROELECTRIC POWER PLANT

Edgar J. Amaya Simeón, edgar.amaya@gmail.com

Alberto José Álvares, alvares@AlvaresTech.com

Rosimarci Pacheco Tonaco, rosimarci@hotmail.com

Rodrigo de Queiroz Souza, rodrigoqsouza@hotmail.com

Universidade de Brasília, Departamento de Engenharia Mecânica e Mecatrônica, Grupo de Inovação em Automação Industrial (GIAI), CEP 70910-900, Brasília, DF, Brasil

Ricardo Ribeiro Gudwin, gudwin@dca.fee.unicamp.br

Universidade Estadual de Campinas (UNICAMP), Faculdade de Engenharia Elétrica e de Computação, Departamento de Engenharia de Computação e Automação Industrial, Campinas, São Paulo, Brasil.

Abstract. *In this paper, we present the conceptualization of an intelligent system to be used on the predictive maintenance of a hydroelectric power plant. It is a real application, being developed in the context of the project "Modernization Processes of Automation Area of the Hydroelectric power stations of Balbina and Samuel", with the support of Eletronorte. The work described in this paper corresponds to the development of an intelligent kernel, which is planned to be the heart of a major methodology, called SIMPREBAL (Predictive Maintenance System of Balbina), to be implanted in the plant of Balbina, near the city of Manaus in Brazil. The overall specification and design of this intelligent kernel is described here using UML (Unified Modeling Language) diagram. This intelligent kernel, called here I-kernel, will be providing expert systems functionality, besides fuzzy logic rules processing and neural networks in an operational cycle where data is collected from equipments and databases, and will be used to both give support to the operational team and to the maintenance team at the plant. I-Kernel will get the data from supervision system by means of both JDBC access to databases and JNI (Java Native Interface) getting data from the OPC (OLE - Object Linking and Embedding - for Process Control) server. In order to provide expert systems and fuzzy logic functionality to the system, we will be using the tools JESS (Java Expert Systems Shell) and Fuzzy-JESS. The SIMPREBAL methodology is based on RCM (Reliability centered maintenance) concepts, being used to analyze the manners and effects of fails on the Hydraulic Generator Units of Balbina, focusing its analysis in the Turbine system. In this work, the main details of the I-Kernel system are detailed. The concept of an intelligent kernel is that will be useful for the construction of similar applications in the future.*

Keywords: *Unified Modeling Language, Predictive Maintenance, Expert Systems, Fieldbus Foundation.*

1. INTRODUCTION

The motivation for this work started with the development of the SIMPREBAL (Alvares, 2006) methodology, under the context of the project "Modernization Processes of Automation Area of the Hydroelectric power stations of Balbina and Samuel", with the support of Eletronorte. The main goal was to implement ideas from RCM (Reliability Centered Maintenance) (Smith, 1993)(Mobray, 1997) and implant them in the plant of Balbina, a hydroelectric power plant located near the city of Manaus in Brazil. The SIMPREBAL project conceived an intelligent system collecting data directly from equipments in the power plant, storing this data in a database, and making use of all this information in order to predict maintenance interventions in order to minimize the impact of (future) failures in the operation of the plant. The main strategy for achieving this goal is by means of the prediction of future failures, based on the measuring of some process variables and analyzing their operational conditions in order to detect anomalous situations that may indicate that a potential failure is possible, in a near future. This may allow for the maintenance team to anticipate this failure and take the appropriate actions in order to minimize the problems that this future failure may impose in the operation of the power plant. In order to develop the SIMPREBAL methodology, there was a need for an intelligent kernel, a software component responsible for getting the data both directly from the equipments used in the power plant, and from databases with historical data, processing all this data in order to analyze them, making predictions, and an integration with the maintenance system of the power plant, so the predicted anomalies may take their role in the organizational process of the maintenance team. From this demand, there appeared the main subject theme of this paper, the I-Kernel Subsystem. We conceived I-Kernel to be a software component, to be developed in Java language, which should be integrated to the main computer system in Balbina, providing the way for implementing SIMPREBAL in the operational (supervision and control) and maintenance organizational process already working in the plant of Balbina.

The role of I-Kernel was, then, to provide the general functionalities of expert system, fuzzy rule based system and neural networks, which were required in order to generate the predictive analysis and decision-making demanded by the SIMPREBAL methodology. In this paper, we detail how we conceived this main component and how it can be integrated to other sub-systems in order to allow for the SIMPREBAL methodology to be applied in a power plant. This will be

DIFFERENT CONTROL STRATEGIES USED IN DIDACTIC PLANT PD-3 OF SMAR THROUGH OPC TECHNOLOGY

Edgar J. Amaya Simeón, edgar.amaya@gmail.com

Alberto José Álvares, alvares@AlvaresTech.com

Victor Rafael R Celestino, cvictor@uol.com.br

Carlos Frederico Maciel Silveira, fredericofurlan@yahoo.com.br

Universidade de Brasília, Departamento de Engenharia Mecânica e Mecatrônica, Grupo de Inovação em Automação Industrial (GIAI), CEP 70910-900, Brasília, DF, Brazil

Abstract. *In this work, three different control strategies, neural network, fuzzy logic and PID (Proportional Integral Derivative), were implemented in Matlab, acting on a Smar PD3 didactic plant by means of OPC (OLE - Object Linking and Embedding - for Process Control) technology. The PD3 plant employs Foundation Fieldbus protocol and configuration tools of Smar System 302. The PD3 process chosen for testing control strategies consists to control the temperature on the mix water tank, which has a constant flow input of constant temperature hot water and an input consisting of ambient temperature water controlled by a pneumatic valve. The PID control was developed using differences equation and adjusting the parameters using a Ziegler- Nichols method. The Neural Network, that has a structure (3-20-1) capable to learn the dynamic of the PID controller, was trained using the values saved in vectors from the response of the PID controller. The training was accomplished using a backpropagation method that is least Square Method (LMS). The LMS method, search the minimum valor of the function square error. The Fuzzy Logic Controller was developed using the toolbox of the Matlab, membership function was edited in the FIS editor. In addition to the controllers a supervisory was developed using GUI of Matlab, acting on PD3 plant independently of SCADA Process View (Smar System 302). Controllers obtained satisfactory results for the combinations of input variables values, the Neural Network and PID controller have similiar response actuation and the Fuzzy Logic Controller has a less accuracy response, in other words the results highlight differences in performance among control strategies.*

Keywords: *Control Systems, Neural Networks, Fuzzy Logic, PID, Fieldbus Foundation.*

1. INTRODUCTION

In this work, the temperature control level in the mix water tank of Smar PD-3 didactic plant was executed, acting through different control strategies, implemented in Matlab. Plant access was achieved using OPC technology (OLE - Object Linking and Embedding - for Process Control). The plant allows the implementation of several control loops, by using industrial devices with sensing, actuating and control functions. These devices are directly managed through a Fieldbus Foundation protocol. PD3 Plant also includes a System 302 software suite.

The objective of this work was to implement and compare three different control strategies, Neural Network, Fuzzy Logic and PID (Proportional Integral Derivative), to communicate with the Fieldbus Foundation Smar instruments located in the PD3 didactic plant, using the Matlab opc toolbox that use OPC technology (OLE - Object Linking and Embedding - for Process Control), also to develop the supervisory system in order to have the HMI (Human Machine Interface).

The rest of this paper is organized as follows: Section 2 presents the description of the PD3 didactic plant; Section 3 presents the three different control strategies and describes the supervisory system; Section 4 describes the tests, results and discussions; and finally Section 5 concludes the paper.

2. SMAR PD3 DIDACTIC PLANT

The Smar PD3 didactic plant aim is to demonstrate didactically the operation of several control loops, using the same equipment and configuration tools (System 302 software suite), developed for the application in industrial control. With a compact arrangement, this plant makes accessible to instructors and students all components in the loop, allowing not only the observation of its behavior, but also its manipulation (Smar, 2004).

In the implementation of such loops, the same characteristics and situations found by instrumentation professionals are provided, with the resources of advanced technology available in the market. Besides the proposed control loops, others can be generated from the same physical structure assembled, without the need of any mechanical modification, but only by modifying the configuration of its devices through functional blocks (Smar, 2005).

The plant applies Fieldbus Foundation technology, an industrial communication bus used to connect field devices such as sensors, actuators, indicators and controllers, which allow the control of temperature, flow and associated levels, helping in industrial instrumentation and control systems learning.

The Smar PD3 didactic plant, presented in the Fig. 1, is monitored and operated by a station, constituted of a PC microcomputer and supervisory software, which executes equipment data acquisition and presents those by screen ani-



8º CONGRESSO IBEROAMERICANO DE ENGENHARIA MECANICA

Cusco, 23 a 25 de Outubro de 2007

SISTEMA INTELIGENTE DE MANUTENÇÃO BASEADA EM CONDIÇÃO PARA USINA HIDRELÉTRICA DE BALBINA

Amaya E. J.*, Tonaco R.*, Souza R. Q.* Álvares A. J.*

*Universidade de Brasília, Departamento de Engenharia Mecânica e Mecatrônica, Grupo de Inovação em Automação Industrial (GIAI), CEP 70910-900, Brasília, DF, Brasil.

*edgar.amaya@gmail.com, *rosimarci@gmail.com, *rodrigoqsouza@gmail.com, *alvares@AlvaresTech.com

RESUMO

Neste artigo é apresentada a metodologia que foi desenvolvida no contexto do projeto de Modernização da Área de Automação de Processos das Usinas Hidrelétricas de Balbina e Samuel, onde o objetivo é o desenvolvimento de um sistema inteligente de manutenção preditiva da usina de Balbina. A metodologia SIMPREBAL é baseada em conceitos de manutenção centrada em confiabilidade (MCC), desenvolvendo um sistema chamado de I-Kernel, um kernel inteligente, que vai obter dados dos bancos de dados do sistema supervisorio da Rockwell, do sistema de gestão da manutenção e operação Maximo, do sistema Assetview e do proprio Simprebal, usando JDBC (Java Database Connectivity) e dados do Servidor OPC (OLE – Object Linking and Embedding – for Process Control) da instrumentação Fieldbus Foundation da Smar usando a biblioteca openscada do projeto Utgard, além de obter dados o I-Kernel processará informações usando sistemas especialistas baseados em regras usando JESS (Java Expert System Shell) e técnicas de Inteligência Artificial tais como, redes neurais e lógica fuzzy usando Fuzzy JESS. Os dados dos defeitos e falhas obtidos vão ser armazenados num banco de dados do Simprebal para serem usados por um sistema de aprendizagem baseado em redes neurais e predizer falhas usando essas informações. O sistema é concebido para coletar e analisar os dados monitorados das unidades geradoras da usina de Balbina, e a implementação de um sistema computacional cliente-servidor com vistas à produção de diagnósticos de estados de funcionamento e de dados que auxiliem a tomada de decisão quanto às ações operacionais e de manutenção das máquinas gerando OS (ordens de serviço) com sugestões das ações de manutenção, visando o aumento da disponibilidade dos equipamentos.

PALAVRAS CHAVE: Manutenção Preditiva, Sistemas Especialistas, Redes Neurais, Lógica Fuzzy, Fieldbus Foundation.

INTRODUÇÃO

O modelo OSA-CBM (Open System Architecture for Condition Based Maintenance) que consiste em sete camadas mostrado na Fig. 1, será utilizado como referência para o desenvolvimento do sistema de manutenção inteligente baseado em condição, maiores detalhes sobre o modelo, podem ser encontrados no site <http://www.mimosa.org/downloads/39/specifications/index.aspx>. A noção de uma arquitetura estendida em camadas usada aqui é consistente com o conceito usado em [Buschman, 1996]. As camadas hierárquicas representam uma transição lógica ou um fluxo da saída dos sensores para a camada de tomada de decisão, através das camadas intermediárias até chegar à camada de apresentação.



Fig. 1: Arquitetura OSA-CBM e suas 7 camadas.

A seguir será apresentado um exemplo das sete camadas estudando o caso do mancal guia da turbina e como estas se ajustam ao sistema SIMPREBAL que é baseado em MCC (Manutenção Centrada em Confiabilidade). Segundo [Fleming et al., 1997], a MCC envolve uma consideração sistemática das funções do sistema, a maneira como essas funções falham é um critério de priorização explícito baseado em fatores econômicos, operacionais e de segurança para a identificação das tarefas de manutenção aplicáveis tecnicamente e custos eficientes no combate a essas falhas.

ESTUDO DE CASO: MANCAL GUIA DA TURBINA

Esta seção apresenta uma breve explicação sobre os mancais do grupo turbina-gerador, bem como, um estudo de caso utilizando como exemplo o mancal guia da turbina. O mancal guia da turbina é responsável por suportar o empuxo radial do eixo da turbina. O arranjo dos mancais do grupo turbina-gerador mostrado na figura 2 é o seguinte: o mancal guia inferior é colocado próximo à extremidade inferior do eixo, na tampa superior e compatível com o arranjo e o acesso à vedação do eixo; o mancal combinado ou mancal guia-escora é suportado por uma estrutura cônica apoiada na tampa superior; e o mancal de guia superior ou mancal do gerador é locado imediatamente acima do rotor do gerador, no centro da aranha superior do gerador, cujo peso deve ser suportado pela carcaça do estator, mas as forças radiais do mancal devem ser suportadas pela estrutura de concreto.

Uma vez definida as funções do sistema utilizando as sete camadas do modelo OSA-CBM, pode-se definir os módulos das camadas para o sistema em estudo do mancal guia da turbina.

Módulo de sensor: O módulo do sensor deve ser desenvolvido de acordo com o padrão IEEE Std 1451 segundo [Bengtsson, 2004]. A aquisição de dados necessários para o processamento da informação relativa ao mancal guia da turbina, a obtenção das Tags referentes a cada equipamento é feita via OPC para Tags da instrumentação Fieldbus Foundation da Smar da serie 302 é feita segundo [Smar, 2001], e as Tags do banco de dados da Rockwell via JDBC mostrado na Fig. 3.

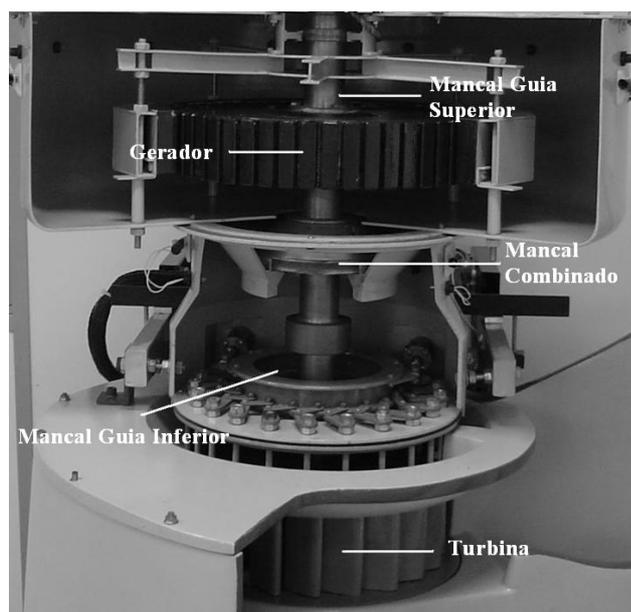


Fig. 2: Maquete de uma Unidade Geradora Hidraulica.

Processamento de sinal: O sistema deve acessar os dados da Usina de Balbina a partir dos Bancos de Dados SQL utilizados pelo Sistema de Monitoramento de Balbina, Sistema de Gestão de Operação e Manutenção Máximo ou diretamente da instrumentação Fieldbus Foundation da Smar através do servidor OPC mostrado na Fig. 3, sendo que estes disponibilizam as informações on-line dos equipamentos. Para obtenção das variáveis on-line via OPC de acordo com as configurações dos blocos funcionais segundo [Smar, 2005], está em desenvolvendo um programa para aquisição das Tags necessárias pra avaliação dos equipamentos usando a biblioteca openscada do projeto Utgard [Utgard, 2007]. Cada Tag tem um valor de Status que é um número de 8 bits, onde os 2 bits mais significativos representam a qualidade do signal do dispositivo (0=Bad, 1=Uncertain, 2=Good Non Cascade, 3=Good Cascade), os 2 bits menos significativos são os limites (0=Not Limited, 1=Low Limited, 2=High Limited, 3=Constant), os quatro bits intermediário representam o substatus, onde para cada valor da qualidade existem valores de substatus e se ocorrer um defeito ou falha é indicado com números diferentes, assim pode-se saber se foi no sensor, equipamento ou outros, segundo a tabela 1. As regras para esta camada mostra-se na tabela 2 onde na primeira coluna compara-se o status e o subStatus com número e na segunda coluna declara-se um código para essa condição da Tag.

Tabela 1: Valores de Sub-status e qualidade no atributo de Status.

		Qualidade			
		0 = Bad	1 = Uncertain	2 = Good (Non Cascade)	3 = Good (Cascade)
Sub-status	0	Não específico	Não específico	Não específica (baixa prioridade)	Não específica
	1	Erro na configuração	Último valor usável	Alarme ativo de bloco	Inicialização aprovada (IA)
	2	Não conectado	Substituto	Alarme ativo de consulta	Requisição de inicialização (IR)
	3	Falha no Equipamento	Valor inicial	Alarme ativo critico	Não solicitado (NI)
	4	Falha no Sensor	Conversão do sensor não exato	Alarme de bloco desconhecido	Não selecionado (NS)
	5	Sem Comunicação, com último valor usavel	Violação de limite de unidade técnica	Alarme de consulta desconhecido	Cancelamento local (LO)
	6	Sem Comunicação, com valor não usável	Sub-normal	Alarme crítico desconhecido	Estado de falha ativo (FSA)
	7	Fora de serviço (alta prioridade)			Estado de falha iniciado (IFS)

Tabela 2: Regras para a camada Processamento de sinal.

Status e subStatus das Tags	Código
status = "2" ou status = "3"	(assert (signal-GOOD ?label))
status = "0" e subStatus = "0"	(assert (signal-BAD-0 ?label))
status == "0" e subStatus = "1"	(assert (signal-BAD-1 ?label))
status == "0" e subStatus = "2"	(assert (signal-BAD-2 ?label))
status == "0" e subStatus = "3"	(assert (signal-BAD-3 ?label))
status == "0" e subStatus = "4"	(assert (signal-BAD-4 ?label))
status = "1" ou (status = "0" e (subStatus = "5" ou subStatus = "6" ou subStatus = "7"))	(assert (signal-UNCERTAIN ?label))

Monitoração de Condição: Esta camada determina a condição do sistema atual, subsistema, ou indicadores de condição de componente baseando-se em algoritmos, sistemas inteligentes e na saída dos módulos de sensores e processamento de sinal. Este módulo de monitoração deve ser desenvolvido usando o padrão ISO 13373-1 segundo [Bengtsson, 2004]. É possível fazer o uso de histórico de condição local e fornecer parâmetros para o modelo. Duas dimensões devem ser consideradas: primeira, o sistema Assetview da Smar e o sistema MES Máximo fornecem análises relativa à condição da instrumentação FieldBus; segunda dimensão, o sistema SIMPREBAL fornece a condição do sistema monitorado. Um exemplo de regras de produção nesta camada para a Tag da pressão de óleo na cuba pode ser visto na tabela 3, onde os valores da tag são comparados com valores da faixa de operação do dispositivo e gera-se um código para a condição do dispositivo.

Tabela 3: Regras para a camada Monitoração de Condição.

Faixa de Valores (mm bar)	Codigo	Condição
(400 < G163MS_AI.PV.VALUE) e signal-GOOD	G163MS-normal	Nível normal de óleo na cuba do mancal guia inferior
(300 < G163MS_AI.PV.VALUE < 400) e signal-GOOD	G163MS-baixo	Nível baixo de óleo na cuba do mancal guia inferior
(200 < G163MS_AI.PV.VALUE < 300) e signal-GOOD	G163MS-alarme	Nível muito baixo de óleo na cuba do mancal guia inferior - 1º estágio
(G163MS_AI.PV.VALUE < 200) e signal-GOOD	G163MS-trip	Nível muito baixo de óleo na cuba do mancal guia inferior - 2º estágio

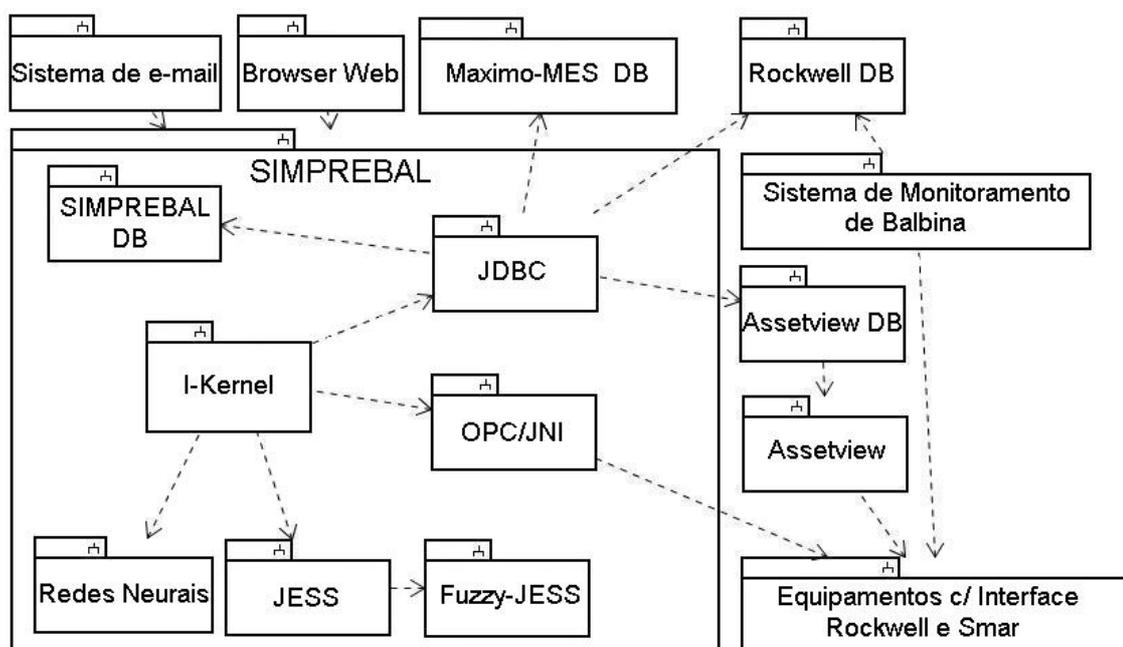


Fig. 3: Visão Geral do Projeto SIMPREBAL, demonstrando o I-Kernel e sua interação.

Avaliação de saúde: Esta camada deve ser desenvolvida a partir das normas IEEE 1232 e a ISO 13373-1 segundo [Bengtsson, 2004]. Na primeira dimensão o conversor FieldBus e o sistema Assetview da Smar fornecem a avaliação da saúde da instrumentação FieldBus. Na segunda dimensão o sistema SIMPREBAL fornece a avaliação da saúde do sistema monitorado. A avaliação da saúde do equipamento ocorre através da extração das características de cada equipamento e posterior detecção de anomalias e diagnósticos dos mesmos. A análise será feita usando os valores históricos das Tags, cálculo do KPI (Key Performance Indicators), os valores das tags serão obtidos do banco de dados do supervisor da Rockwell. No encerramento das OS (Ordens de Serviço), o usuário deverá encerrar as OS fornecendo todos os dados para registro histórico e cálculo de KPI.

Prognósticos: Na tarefa de prognóstico, considera-se a avaliação da saúde do sistema, subsistemas, componentes, o escalonamento empregado (predição de uso) e a capacidade de raciocínio do modelo para prever o estado de saúde do equipamento. Um prognóstico é sugerido a partir das informações dos módulos anteriores. Este pode ser baseado em modelos pré-estabelecidos ou feito considerando as características encontradas na avaliação de saúde. O prognóstico vai consistir de dois passos, o diagnóstico que será feita com os valores atuais e o prognóstico que vai ser feito, usando tendência de tags armazenadas no banco de dados ou através de redes neurais que vão prever falhas ou defeitos futuros baseadas em dados históricos ou acontecimentos passados. As regras para diagnóstico são baseadas em valores reais e são mostradas na tabela 4.

Tabela 4: Regras de produção para Diagnóstico.

Sinal de "g1.st.p.oleo.cuba"	Codigo de Diagnostico
signal-BAD-0	assert(diagnostic G163MS-signal0)
signal-BAD-1	assert(diagnostic G163MS-signal1)
signal-BAD-2	assert(diagnostic G163MS-signal2)
signal-BAD-3	assert(diagnostic G163MS-signal3)
signal-BAD-4	assert(diagnostic G163MS-signal4)
signal-UNCERTAIN	assert(diagnostic G163MS-signal5)
not (condition G163MS-normal)	assert(diagnostic G163MSF1)

Tomada de decisão: Tem o objetivo de integrar informações necessárias para a tomada de decisão, considerando informações sobre a saúde dos equipamentos e para prever a saúde de um sistema e/ou subsistema. O sistema SIMPREBAL irá realizar a tomada de decisão baseando-se na sua base de conhecimento gerada a partir da árvore de faltas/falhas, da árvore de sintomas, e pelas informações de inspeções realizadas pelo sistema TPM (Manutenção Produtiva Total), segundo [Wyrebski, 1998] hoje é considerada uma metodologia que auxilia na redução de ocorrência de falhas em instalações de máquinas e equipamentos nas empresas. Nesse módulo o sistema inteligente apresenta algumas sugestões para a solução do problema, cabe ao usuário tomar a decisão mais aplicada. As regras para esta camada com as sugestões de manutenção são mostradas na tabela 5, a decisão é enviada do servidor i-kernel para o cliente onde é mostrada na tela que interage com o pessoal da operação, gerando OS para manutenção.

Tabela 5: Regras de produção para Tomada de Decisão.

Codigo	Decisão
G163MS-signalD0	Falha no processamento de sinal do tag G163MS: Não específica
G163MS-signalD1	Falha no processamento de sinal do tag G163MS: Erro na configuração
G163MS-signalD2	Falha no processamento de sinal: verificar conexão do sensor de pressão à cuba e ao transmissor G163MS
G163MS-signalD3	Verificar falha no transmissor de pressão G163MS
G163MS-signalD4	Verificar falha no sensor de pressão diferencial da cuba do mancal guia inferior
G163MS-signalD5	Verificar falha de comunicação na aquisição do tag G163MS
G163MSD1	Verificar estanqueidade das tubulações, válvulas e tampas de acesso
	Verificar aperto dos parafusos de fixação da tampa e suporte da cuba
	Realizar alinhamento e ajuste de folga das sapatas
	Restituir a carga de óleo

Apresentação: Suporta a apresentação das informações para o controle das entradas dos usuários do sistema (por exemplo, manutenção e operação). As saídas incluirão qualquer informação produzida pelas camadas mais baixas e as entradas incluirão qualquer informação requerida pelas camadas mais baixas. A figura 4 apresenta a interface homem-máquina associada ao lado do Cliente, sendo desenvolvida em Java para ser usada como uma aplicação Web ou um Applet chamado via browser, independentemente da plataforma computacional usada pelo cliente. A GUI (Interface Gráfica com o Usuário) mostrada na Fig. 4 apresenta a camada de Apresentação (sétima camada) do sistema Simprebal mostrando a Tomada de Decisão (sexta camada) sugerida pelo sistema SIMPREBAL. A GUI apresenta também os valores das Tags em inspeção de variáveis e a estrutura hierárquica dos ativos.

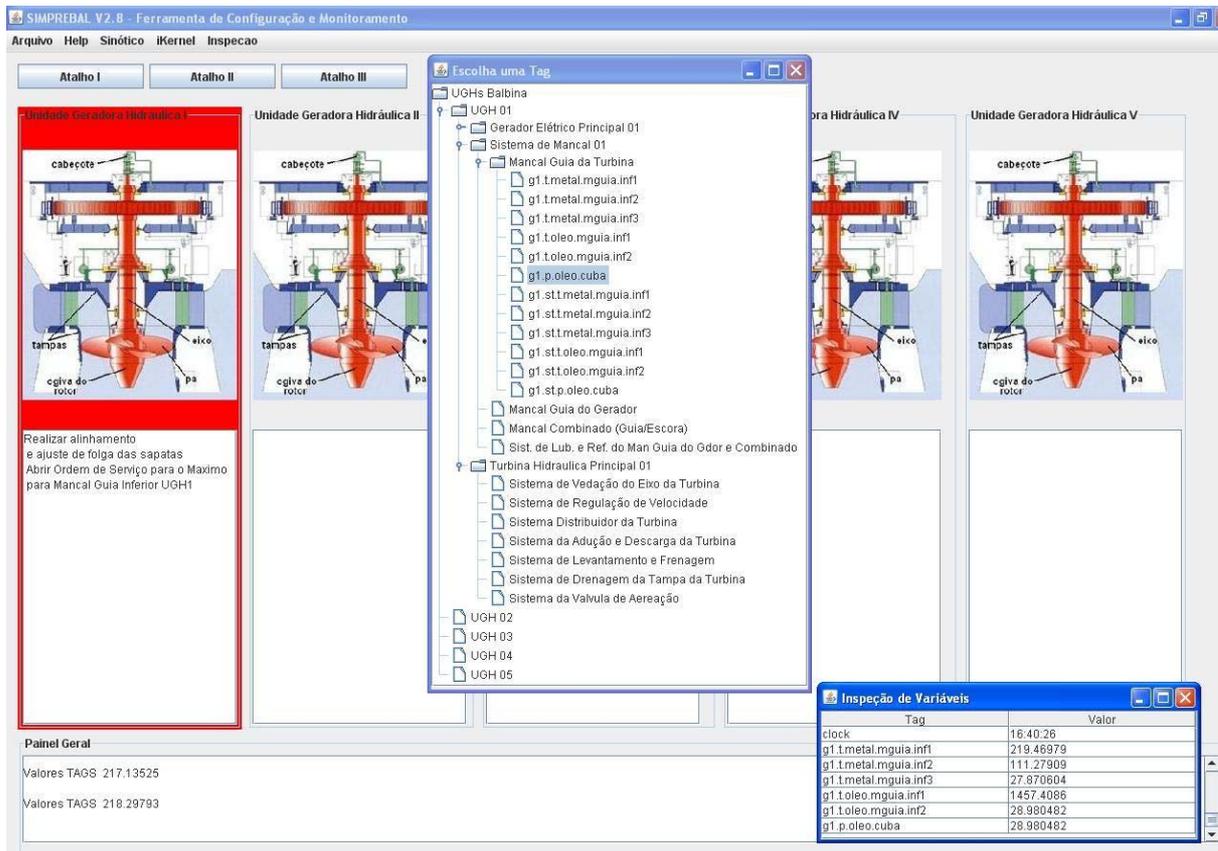


Fig. 4: GUI SIMPREBAL Configuração e Monitoração: Cliente Web referente à camada de Apresentação da metodologia SIMPREBAL.

CONCLUSÃO

Com a proposta apresentada no presente artigo, espera-se mostrar sugestões de manutenção para os operadores e o pessoal de manutenção antes que o equipamento apresente problema. Esse processo de manutenção é conhecido como manutenção preditiva e é baseada em confiabilidade. O sistema SIMPREBAL será desenvolvido usando técnicas de IA (Inteligência Artificial) como regras de produção, Lógica Fuzzy e redes neurais, o que facilitará o processo de tomada de decisão, visto que um dos objetivos do sistema é avaliar e monitorar o equipamento para oferecer prognósticos confiáveis. Permitindo dessa maneira que o operador possa tomar decisões de manutenção com base em informações geradas pelo sistema. O Sistema Inteligente SIMPREBAL auxiliará na tomada de decisão sugerindo prognósticos, para que a manutenção do equipamento possa ocorrer antes que a falha ocorra. Isto implica em ganho de tempo e maior confiabilidade do sistema da usina. Vale ressaltar que a metodologia aqui proposta ainda está em fase de elaboração, resultados concretos serão obtidos posteriormente.

AGRADECIMENTOS

Agradecemos à Eletronorte, Manaus Energia e ao Eng. Antonio Araújo da Eletronorte pelas informações cedidas que foram de extrema importância para o desenvolvimento do presente trabalho.

REFERÊNCIAS

1. F. Buschman, *Oriented Software Architecture A System of Patterns*, England, 1996.
2. OSA-CBM, *Open Systems Architecture for Condition-Based Maintenance*.
<<http://www.mimoso.org/downloads/39/specifications/index.aspx>>, acessado em agosto de 2007.
3. P. V. Fleming e S.R.R.O. França, *Considerações Sobre a Implementação Conjunta de TPM e MCC na Indústria de Processos*, Abraman 12º Congresso Brasileiro de Manutenção: TT044, São Paulo, pp.53, 1997.
4. M. Bengtsson, *Condition Based Maintenance System Technology –Where is Development Heading?*, *Proceedings of the 17th European Maintenance Congress*, Spain, 2004.
5. Smar, *Equipamentos de Campo série 302 Foundation. Manual de instalação, operação e manutenção*, 2001.
6. Smar, *Manual de instruções dos blocos funcionais Fieldbus Foundation*”, 2005.
7. Utgard, *Projeto Utgard*, <<http://openscada.org/UtgardProject>>, acessado em agosto de 2007.
8. J. Wyrebski, *Manutenção Produtiva Total um Modelo Adaptado*”, *Dissertação (Mestrado em Engenharia de Produção) Programa de Pós-Graduação em Engenharia de Produção, UFSC*, 1998.

Edgar Jhonny Amaya Siméon – Aluno do curso de Sistemas Mecatrônicos – UnB

Prof. Alberto José Álvares – ENM/UnB (Orientador)