

DESENVOLVIMENTO DE UM FRAMEWORK DE NÍVEL 2 PARA APLICAÇÕES DE AUTOMAÇÃO INDUSTRIAL¹

*Gilcésar Geraldo Sebastian Nogueira de Ávila²
Tales Wallace Souza³*

Resumo

A Aperam é uma empresa bastante informatizada, dispendo sistemas em toda a pirâmide do CIM (Nível 5 – SAP, Nível 4 – PCP, Nível 3 – MES, Nível 2 – Otimizadores, Modelos Matemáticos e Receitas, Banco de Dados Temporais e Relacionais, Nível 1 – Sistemas de Supervisão, Nível 0 – PLC, Drivers e Instrumentação). Para evitar a diversificação dos sistemas de Nível 2 e por não existir ainda um sistema de Nível 2 que contempla as necessidades da Aperam (forte integração com o Nível 1 e com o MES da Aperam – sistema proprietário e customizado), a equipe de automação decidiu desenvolver um Framework de Nível 2. O Framework é um pacote de software básico, feito em Java, que fornece as APIs necessárias para o acesso aos recursos em um sistema de automação. Assim um aplicativo de Nível 2 poderá acessar os demais níveis de automação, Nível 1 e Nível 3, com grande abstração sem se preocupar com detalhes de implementação destas API's. Como o acesso a tais recursos já estão disponibilizados, ocorre por consequência um ganho no tempo de desenvolvimento dos aplicativos de Nível 2, além de aumentar a confiabilidade dos mesmos, pois tais APIs já foram massivamente testadas.

Palavras-chave: Automação industrial; Framework; Padronização.

DEVELOPMENT OF A LEVEL 2 FRAMEWORK FOR INDUSTRIAL AUTOMATION APPLICATIONS

Abstract

The Aperam company is a very computerized, providing systems throughout the CIM pyramid (Level 5 - SAP, Level 4 - CFP, Level 3 - MES Level 2 - Analysis and Optimization, Mathematical Models and Revenue, and Temporal Database Relational Level 1 - Oversight Systems, Level 0 - PLC Drivers and Instrumentation). To avoid the diversification of Level 2 systems and there is still no system of Level 2 which includes the needs of Aperam (tight integration with Level 1 and the MES Aperam - proprietary and custom), the automation team decided develop a Framework Level 2. The Framework is a basic software package, written in Java, which provides the necessary APIs for access to resources in an automation system. Thus an application can access Level 2 other levels of automation, Level 1 and Level 3, with great abstraction without worrying about implementation details of these APIs. As the access to such resources are already available, there is consequently a gain in time of development of Level 2 applications, and increase their reliability, as these APIs have been massively tested.

Key words: Industrial automation; Framework; Patterning.

¹ *Contribuição técnica ao 16º Seminário de Automação e TI Industrial, 18 a 21 de setembro de 2012, Belo Horizonte, MG.*

² *Analista de Sistemas, Analista de automação da gerência de tecnologia de automação industrial da APERAM em Timoteo – MG.*

³ *Analista de Sistemas, Analista de automação da gerência de tecnologia de automação industrial da APERAM em Timoteo – MG.*

1 INTRODUÇÃO

Nos sistemas de Nível 2 da Aperam encontram-se os sistemas desenvolvidos com o objetivo de otimizar a produção. Estes sistemas utilizam recursos como: modelos matemáticos, sistemas especialistas, lógicas nebulosas, redes neurais, etc. Neste nível estão presentes, também, as ordens de produção e emissão de relatórios. Este nível é responsável também pela interface de comunicação entre os sistemas de Nível 3 (Planejamento da Produção) e de Nível 1 (Controle e Supervisão).

Ao se comprar uma nova linha de produção ou em um revamping, os fornecedores eletro-mecânicos, normalmente, também forneciam soluções de automação que na maioria das vezes se mostravam inadequadas para a realidade de nossa empresa, estes sistemas eram distintos (mesmo para um mesmo fornecedor), caros, de difícil manutenção e portabilidade, de diversas arquiteturas, sistemas operacionais e linguagens, com interface homem-máquina não amigável e distante da realidade da nossa empresa.

Assim, os Sistemas de Nível 2 da Aperam eram desenvolvidos utilizando a tecnologia que o fornecedor dominava, desde a linguagem de programação até a arquitetura de software das mais diversas formas, isto causou um aumento tanto da diversidade quanto da complexidade de soluções, a manutenção se tornava cada vez mais cara e demorada, as interferências na produção estavam muito elevadas devido à dificuldade de localização de erros nos sistemas, ou mesmo erros causados pela própria equipe devido à falta de domínio do software. Isto demandava vasta mão de obra especializada e elevado tempo na manutenção e melhoria dos sistemas, devido à grande variedade de linguagens de programação, arquiteturas de software etc.

Isto juntamente com o constante aumento da competitividade entre as indústrias siderúrgicas e aliado à necessidade de manter equipes cada vez mais otimizadas, fazia-se necessário, primeiramente, diminuir a interferência na produção e por conseqüência diminuir a complexidade dos Sistemas de Nível 2. Para alcançar estas metas seria necessário padronizar o desenvolvimento de Aplicações de Nível 2, além de diminuir o tempo de projeto, de manutenção e de evolução dos sistemas.

Deste cenário surgiu a primeira versão do framework desenvolvido no formato Cliente/Servidor utilizando como parte do servidor a linguagem Java e na parte do cliente o Delphi. Isto permitiu iniciar a jornada para a padronização dos Sistemas de Nível 2 da Aperam.

O Framework de Nível 2 da Aperam foi concebido na tentativa de limitar alguns destes problemas no desenvolvimento de sistemas de Nível 2. Tais problemas acabam por gerar custos excessivos em todo o ciclo de vida do produto.

Dotado de ferramentas de acesso aos recursos do sistema, ele permite um ganho de produtividade no desenvolvimento e manutenção dos aplicativos de Nível 2. Possui um cliente OPC capaz de se conectar à maioria dos servidores OPC do mercado, possui um sistema de auditoria, que gera alarmes e notifica eventos decorrentes de ações operacionais, estados anormais do sistema, exceções, etc. Possui arquitetura Cliente/Servidor, sendo o cliente desenvolvido em .NET e o servidor em Java. Permite o desenvolvimento de aplicações baseados em técnicas modernas de desenvolvimento de software ágil como o RUP e OpenUp.

Uma de suas características é a grande facilidade de comunicação com sistemas legados e outros sistemas, desde que esses tenham uma interface conhecida, por exemplo, é possível estabelecer comunicação com um ESB; com balanças Toledo, Alpha; módulos ADAM, etc. Utiliza uma camada de persistência de dados que

permite acessar a maioria dos banco de dados de dados do mercado. O Framework é um produto que exibe muita flexibilidade, escalabilidade e liberdade no desenvolvimento de sistemas de Nível 2.

Frameworks possuem vantagens, tais como: maior facilidade para a detecção de erros, por serem peças mais concisas de software; concentração na abstração de soluções do problema que estamos tratando; eficiência na resolução dos problemas e otimização de recursos.

O framework atua onde há funcionalidades em comum a várias aplicações, porém para isso as aplicações devem ter algo razoavelmente grande em comum para que o mesmo possa ser utilizado em várias aplicações.

Segundo Fayad e Schmidt: “Framework é um conjunto de classes que colaboram para realizar uma responsabilidade para um domínio de um subsistema da aplicação.”

Segundo definição do *Wikipedia*, em desenvolvimento de software, um Framework é “uma abstração que une códigos comuns entre vários projetos de software provendo uma funcionalidade genérica. Um framework pode atingir uma funcionalidade específica, por configuração, durante a programação de uma aplicação.

“Ao contrário das bibliotecas, é o framework quem dita o fluxo de controle da aplicação, chamado de Inversão de Controle.”

Framework conceitual é um conjunto de conceitos utilizados para resolver um problema de um domínio específico. Framework conceitual não se trata de um software executável, mas sim de um modelo de dados para um domínio. Framework de software compreende um conjunto de classes implementadas em uma linguagem de programação específica, usadas para auxiliar o desenvolvimento de software. Especificamente em orientação a objetos, framework é um conjunto de classes com objetivo de reutilização de arquitetura de software, provendo um guia para uma solução em um domínio específico de software. Framework se diferencia de uma simples biblioteca, pois esta se concentra apenas em oferecer implementação de funcionalidades, sem definir a reutilização de uma solução de arquitetura. Muitos engenheiros acreditam que a arquitetura é determinada pelos requisitos e por isso esperam que a fase de engenharia dos requisitos esteja finalizada para então iniciar sua arquitetura. Porém, apenas uma fração dos requisitos específicos do sistema tem influência na arquitetura. A identificação dos requisitos que são significantes para a arquitetura pode ser respondida através de um framework conceitual desenvolvido especialmente para um domínio específico, uma vez que esta resposta é muito dependente do domínio. Avançar para a fase de projeto ou mesmo iniciar a implementação do sistema não quer dizer que a definição da arquitetura esteja finalizada. Isto significa que o detalhamento obtido até então já é suficiente para prosseguir com o projeto de uma parte do sistema.

Frozenspots são as partes fixas de um framework, também conhecidos como hook points. São serviços já implementados pelo framework. Normalmente realizam chamadas indiretas aos hotspots.

Hotspots são as partes flexíveis de um framework. São pontos extensíveis, necessitam de complementação por funcionalidades/serviços que devem ser implementados. Hotspots são partes nos quais os programadores que usam o framework adicionam o seu código para especificar uma funcionalidade de sua aplicação. São invocados pelo framework ou recebem mensagens de uma classe do framework (frozenspot). Isso geralmente é implementado através de polimorfismo: heranças, interfaces e métodos abstratos;

2 MATERIAIS E MÉTODOS

O Framework de Nível 2 define que os aplicativos de Nível 2 seguirão o padrão de projetos MVVM.

O Framework é um conjunto de funções que fornece acesso aos recursos do sistema e às funcionalidades que sejam comuns aos sistemas de Nível 2. Foram utilizadas tecnologias disponíveis no mercado para a construção do Framework. A engenharia de software foi utilizada para gerar um produto que reutilizasse algumas tecnologias disponíveis no mercado, estas foram agrupadas e ajustadas de acordo com a necessidade da empresa, ao final foi obtido um produto com pouca codificação própria e grande reutilização de códigos de terceiros, dentre as funções disponibilizadas pelo framework podemos descrever as que seguem:

2.1 Comunicação com CLP e Módulos Via – Tecnologia OPC

Um dos grandes problemas de se interfacear equipamentos e sistemas no chão de fábrica reside em se compatibilizar os protocolos da camada de aplicação. O principal objetivo do protocolo OPC (OLE for Process Control) é definir uma maneira única de comunicação, facilitando assim a implementação de mecanismos de escrita e leitura entre sistemas e equipamentos de campo

O cliente OPC fornece funções para acessar um servidor OPC. Para acessar um servidor OPC, foram implementadas interfaces que definem as operações definidas na especificação do OPC Foundation. Para tal foi utilizado um pacote de classes denominado JEASYOPC, cuja licença é GPL e o código fonte pode ser obtido no site do desenvolvedor.

2.2 Barramento de Comunicação com outros Sistemas – Tecnologia ESB

O Framework provê comunicação com outros sistemas através de um barramento ESB. Tais comunicações podem ser feitas para qualquer sistema ligado a este barramento, normalmente este canal de comunicação é utilizado para troca de informações com o Nível 3, embora possa ser utilizada também para comunicação entre Sistemas de Nível 2.

A própria interface se encarrega de manter a consistência das informações, não permitindo que valores fora das faixas especificadas sejam enviados e recebidos, aumentando a robustez do sistema e a confiabilidade das informações.

2.3 Acesso a Dados – Tecnologia Hibernate e JPA

Uma biblioteca baseada no *JPA (Java Persistent API)* foi disponibilizada para permitir o acesso aos dados persistentes fornecendo métodos simplificados para a armazenagem e recuperação dos dados de uma base de dados. Esta biblioteca utiliza o *Hibernate*, que é um *framework* de persistência de dados escrito em Java, que abstrai a utilização de banco de dados relacional em sistemas orientado a objetos. Uma das características deste método é a liberdade de se escolher praticamente qualquer sistema de armazenamento de dados, como Oracle, Postgre, MySQL, SQLServer, etc.

2.4 Sistema de Auditoria – Tecnologia Log4j

O pacote de auditoria (Log) é parte de grande importância em um sistema de automação, pois, é ele que fornece recursos para análise de falhas nos sistemas e faz, adicionalmente, a gestão de eventos e alarmes. Todo o sistema de auditoria é configurável, assim, é possível alterar uma mensagem de alarme, suprimi-la ou mesmo modificar sua prioridade, sem necessidade de recompilar o sistema.

2.5 Comunicação entre os Processos – Tecnologia RMI

Um sistema de Nível 2 é dividido em processos que se comunicam uns com os outros, formando um rede cooperativa. Um processo é uma classe responsável por resolver um problema específico dentro da aplicação, possuindo uma arquitetura baseada na tecnologia *RMI*, cada um destes processos é executado exclusivamente em *uma Máquina Virtual Java (JVM)*, permitindo assim uma grande escalabilidade e robustez. Quando um processo, eventualmente, apresenta um problema ele pode ser reiniciado sem afetar o funcionamento dos demais, da mesma forma que pode ser também atualizado.

2.6 Pacote Gestão

Este pacote fornece funcionalidades comuns aos os sistemas Nível 2 da Aperam, estas funcionalidades foram identificadas e foram desenvolvidas interfaces padronizadas com o Nível 3. Existindo tal padrão foi possível criar os módulos e reaproveitá-los em qualquer equipamento da usina sem nenhum esforço de programação, apenas parametrização. Cada um destes módulos podem ser habilitados ou desabilitados, dependendo da especificação do projeto.

2.6.1 Gestão de paradas e setups

Este módulo controla o tempo de equipamento parado ou funcionando. Um equipamento está parado quando atende a requisitos previamente definidos, por exemplo: velocidade abaixo de 5 m/min, ou bit do PLC indicando equipamento parado. O *setup* é uma parada que é prevista no processo e faz parte do mesmo e, portanto, não é contabilizado como parada. A gestão de paradas controla precisamente o quanto um equipamento efetivamente trabalhou.

	Início	Fim	Código	Cons.	Compl.	Natur.	Equipe	Turno	Duração Prevista
	18/05/2012 15:10:21	18/05/2012 15:21:53	9998	OI	0	OP	IN	3	
	18/05/2012 14:56:37	18/05/2012 15:10:17	9998	OI	0	OP	IN	3	
	18/05/2012 13:42:13	18/05/2012 14:26:36	9998	OI	0	OP	IN	2	
	18/05/2012 12:45:25	18/05/2012 12:58:25	9998	OI	0	OP	IN	2	
	18/05/2012 11:57:37	18/05/2012 12:40:51	9998	OI	0	OP	IN	2	
	18/05/2012 11:31:51	18/05/2012 11:46:53	9998	OI	0	OP	IN	2	
	18/05/2012 10:33:40	18/05/2012 11:07:07	9998	OI	0	OP	IN	2	
	18/05/2012 09:19:22	18/05/2012 09:52:25	9998	OI	0	OP	IN	2	
	18/05/2012 08:27:47	18/05/2012 08:56:39	9998	OI	0	OP	IN	2	
	18/05/2012 07:56:49	18/05/2012 08:11:56	9998	OI	0	OP	IN	2	
	18/05/2012 07:34:10	18/05/2012 07:54:51	9998	OI	0	OP	IN	2	
	18/05/2012 07:23:10	18/05/2012 07:34:09	9998	OI	0	OP	IN	2	
	18/05/2012 06:00:14	18/05/2012 07:22:41	9998	OI	0	OP	IN	1	
	18/05/2012 03:39:13	18/05/2012 05:30:19	9998	OI	0	OP	IN	1	
	18/05/2012 02:48:03	18/05/2012 03:11:14	9998	OI	0	OP	IN	1	
	18/05/2012 01:50:41	18/05/2012 02:25:11	9998	OI	0	OP	IN	1	
	18/05/2012 01:04:31	18/05/2012 01:32:23	9998	OI	0	OP	IN	1	
	18/05/2012 00:35:29	18/05/2012 00:53:27	9998	OI	0	OP	IN	1	
	18/05/2012 00:10:55	18/05/2012 00:21:27	9998	OI	0	OP	IN	1	
	17/05/2012 22:20:40	17/05/2012 23:52:50	9998	OI	0	OP	IN	3	
	17/05/2012 21:50:21	17/05/2012 22:05:30	9998	OI	0	OP	IN	3	
	17/05/2012 21:23:07	17/05/2012 21:50:19	9998	OI	0	OP	IN	3	
	17/05/2012 20:43:05	17/05/2012 21:01:40	9998	OI	0	OP	IN	3	
	17/05/2012 20:10:58	17/05/2012 20:42:29	9998	OI	0	OP	IN	3	
	17/05/2012 19:22:12	17/05/2012 19:36:29	9998	OI	0	OP	IN	3	
	17/05/2012 17:59:25	17/05/2012 18:53:55	9998	OI	0	OP	IN	3	
	17/05/2012 17:17:20	17/05/2012 17:35:54	9998	OI	0	OP	IN	3	
	17/05/2012 16:33:59	17/05/2012 16:52:35	9998	OI	0	OP	IN	3	

Figura 1 - Gestão de paradas e setups.

2.6.2 Gestão de metrologia

A Gestão de Metrologia cuida dos registros de calibração dos instrumentos utilizados na linha de produção. O mantenedor responsável pela calibração do instrumento informa ao Nível 2 a data da última calibração e a data da próxima calibração. De posse destes dados o Nível 2 emite alarmes de vencimento de calibração e indica o estado atual do instrumento (conforme, não conforme ou uso limitado). O estado também é enviado ao sistema de Nível 1 para indicação nas telas de operação do sistema supervisório.

2.6.3 Gestão de recursos

A função Gestão de Recursos é responsável pela gestão da vida útil dos diversos recursos montados na linha de produção. Como exemplos de recursos a serem monitorados têm: rolos, tesouras, cilindros de laminação, etc. Os recursos são definidos através de interfaces no Nível 3 e informados ao sistema de Nível 2 através de mensagens de montagem e desmontagem ou feita no próprio Nível 2. A contagem da vida útil é realizada pelo Nível 1 tendo como base os parâmetros do recurso definidos pelo sistema de Nível 3. Alarmes e indicações de estado de recurso são gerados no Nível 2 para aviso do operador e registro de não conformidades.

2.6.4 Gestão de acesso

É um mecanismo para controle de acesso ao sistema. Baseado na premissa de que para cada função do sistema existe um ou mais grupos de usuários com acesso à mesma. Assim um usuário pode pertencer a um ou mais grupos de forma a controlar o seu nível de acesso ao sistema.

2.6.5 Gestão de insumos

O Framework disponibiliza interfaces padronizadas para o controle de insumos. Insumo é tudo aquilo que é consumido ou transformado no decorrer do processo do equipamento, por exemplo: Energia elétrica e óleo.

2.7 Processo Orientado a Eventos

Ao ser iniciado, cada processo, aguarda a chegada de algum evento para então disparar a tarefa de tratamento do evento recebido em seguida ele volta a esperar que um próximo evento chegue.

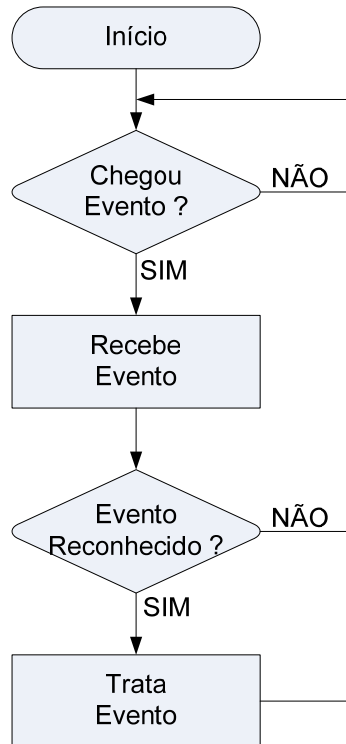


Figura 2 - Fluxograma - Processo Orientado a Eventos.

2.8 Redundância e Processo Monitor

O processo Monitor é o módulo responsável por controlar todo o ciclo de vida dos processos da aplicação. É ele o responsável pela redundância do sistema, isto é, em caso de falha em um computador servidor ele automaticamente alterna a execução do aplicativo para o outro servidor.

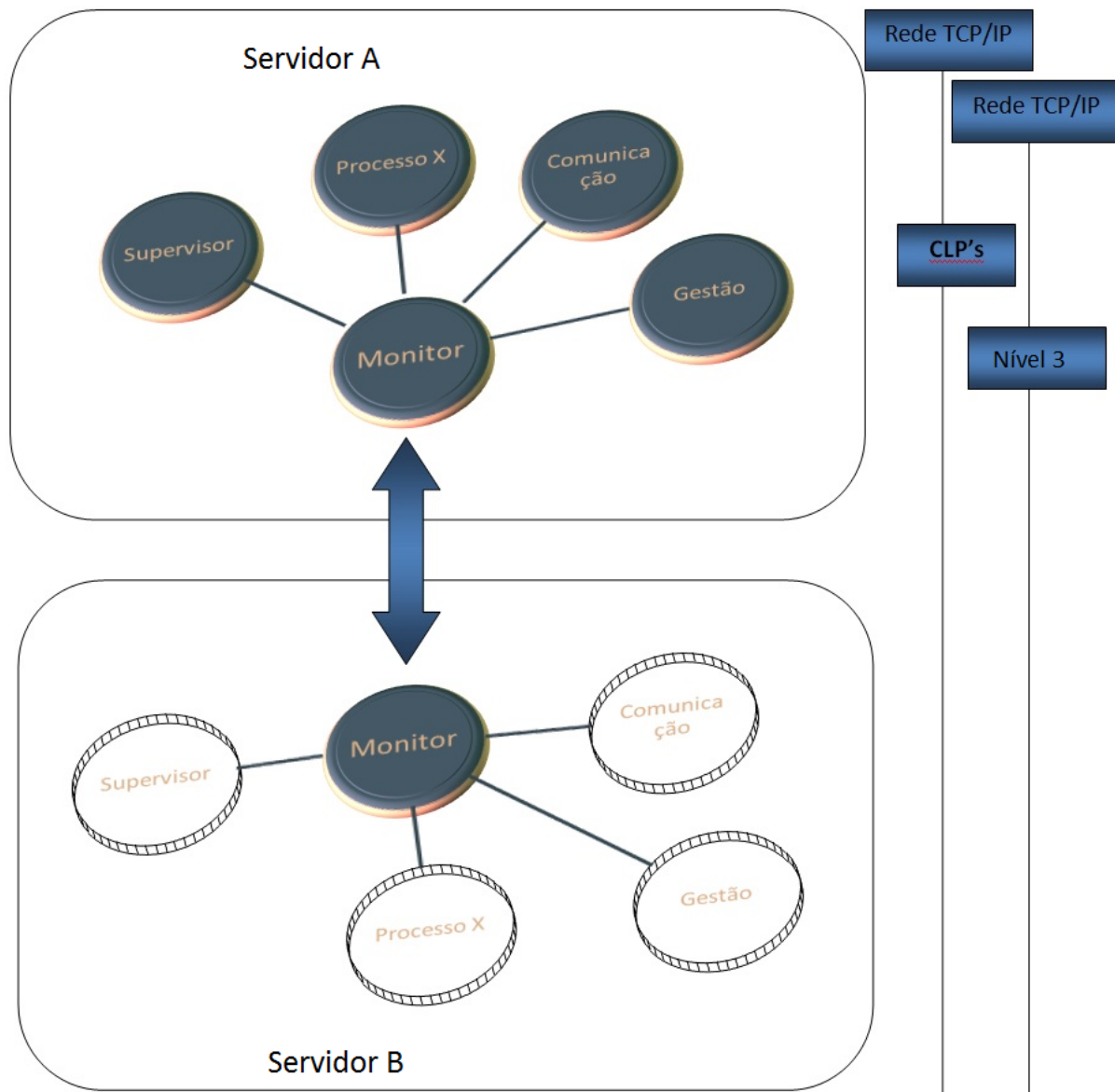


Figura 3 - Redundância e Processo Monitor.

3 RESULTADOS

3.1 Desenvolvimento de Sistemas de Nível 2 Utilizando o Framework de Nível 2

Desde sua primeira versão o Framework de Nível 2 vem mostrando resultados cada vez mais promissores para o desenvolvimento de aplicações de Nível 2. Atualmente um total de treze sistemas, que utilizam o Framework, já se encontram em operação, instalados nas áreas da Aciaria, Redução e Laminação a Frio. Estes sistemas, no geral, se mostraram bastante estáveis, diminuindo os tempos de interferência na produção relativos aos sistemas anteriormente utilizados. O custo de desenvolvimento e manutenção de cada sistema também baixou.

Em média o custo de desenvolvimento de um sistema de Nível 2 utilizando o Framework é cerca de 50% de um sistema que tenha que ser desenvolvido sem o Framework. Esta redução de custos é devido, principalmente, a: Componentes reutilizáveis disponíveis; Funcionalidades comuns já prontas; Utilização de tecnologias bem difundidas; Vasta mão de obra especializada em Java e .NET; Padronização; Curva de aprendizagem na sua utilização já está estabilizada;

S	UM	Inst.	Prazo Conf.	Aço	Larg. (mm)	Esp. (mm)	Peso (Kg)	Comp. (m)	Diâm. Int. (mm)	Pedido	Cliente	Op. Prog.	Origem	Destino
	202744I3000B	X	20/5/2012	P441A	1240,0	1,500	15955	1121	X	X		CO001	RB1	
	241699H3000B	X	31/5/2012	P304A	1240,0	1,750	16910	966	X	X		CO001	LE1	
	241751G8000B	X	30/4/2012	P430E	1290,0	0,400	16885	4338	508	X		CO001	RB4	
	202409I2000B	X	10/5/2012	P439A	1240,0	0,500	24075	5106	X	X		CO001	LE1	
	202769F8000B	X	31/5/2012	P409H	1040,0	2,150	13415	769	X	X		CO001	RB1	
	202739I9000B	X	31/5/2012	P409A	1250,0	2,150	22715	1105	X	X		CO001	RB1	

Figura 4 – Tela padrão do Framework, Programação da produção.

Equip.	Resp.	Cod. Defeito	Intens.	Comp. Ini.	Comp. Fim	Larg. Ini.	Larg. Fim	Face	Int. Rep.	Fim Rep.
LB3	029	B	15	47	0,0	1240,0	D			
LC2	034	A	15	1121	0,0	1240,0	D			
LC2	090	A	15	1121	0,0	1240,0	D			
RB1	250	A	15	1121	0,0	1240,0	D			
RB1	076	A	15	1121	0,0	1240,0	D			

Figura 5 - Tela padrão do Framework, Mapa de defeitos.

4 DISCUSSÃO

O Framework não se limita a comunicação OPC apenas, é possível utilizar drivers nativos de comunicação com CLP's, desde que a interface deste driver seja conhecida. Outro recurso também é a opção de desenvolver os próprios drivers,

como exemplo, um driver que foi desenvolvido para comunicação com RMC's da ABB utilizando o protocolo VIP da própria ABB. Este driver foi desenvolvido e facilmente integrado ao Framework através das interfaces disponibilizadas para este fim.

A manutenção do aplicativo servidor se tornou muito mais simples, pois, é possível alterar o código fonte, recompilar e rodar a alteração sem necessidade de reinicialização do aplicativo, isto porque o carregamento das classes Java é feito dinamicamente, assim quando um arquivo de classe é alterado, da próxima vez que este for utilizado ele será recarregado na memória de forma dinâmica. Isto elimina a necessidade de aguardar oportunidades de parada de produção do equipamento para implantar novas funcionalidades, ou para correção de *bugs*.

No Laminador de Bobinas 1 da Aperam um Nível 2 baseado no Framework substitui um sistema supervisorio feito em FactoryLink 6.6. Apesar de usar OPC, o sistema coleta dados bufferizados no CLP, em um tempo < 40 ms.

A discussão é a parte principal do trabalho. Deve comparar os resultados obtidos pelo autor, com o resultado obtido por outros autores, estabelecendo relações, deduções paralelas.

5 CONCLUSÃO

Os avanços contínuos da tecnologia de TI permitem cada vez mais a integração entre TI e TA. Hoje muitos dos recursos que eram exclusivamente utilizados em TI já são largamente utilizados em automação industrial. Novos horizontes vão se abrindo quando se pensa em virtualização e computação em nuvens.

Hoje a Aperam ainda possui uma diversidade de sistemas instalados das mais diversas tecnologias, tais sistemas demandam alta demanda da equipe de automação, devido à sua complexidade. Alguns destes sistemas serão atualizados para o padrão do Framework.

Quadro 1 – Relação sistemas fora do padrão do Framework

Fornecedor	Sistema Operacional	Linguagem	Quant. Sistemas
ABB	Windows	C#	3
	Unix	C / Oracle Forms	2
	Windows	C / VB	1
VAI	VMS	C / VB	1
	Windows	C	1
Stein	VMS / Windows	Factory Link / C / Fortran	1
Danieli	Windows	Delphi	1
	Unix	C	1
ATAN	Windows	Scada Factory Link	2
	Windows	Delphi	7
Contraste	Windows	Scada Factory Link	1