

DESMISTIFICANDO AS INTERFACES DE AUTOMAÇÃO COM OS SISTEMAS CORPORATIVOS/ERPs¹

Wilson Laizo Filho²
Wanderson Alvarenga Silveira³
Bruno Aguiar do Prado⁴
Renzo Bottino Ziegler⁵
Thiago Oliveira⁶

Resumo

Em todos os projetos de MES desenvolvidos pela Schneider Electric, um ponto que sempre se destaca e causa arrepios tanto nas equipes que implementam os sistemas como nos clientes são as interfaces com os sistemas corporativos. Este trabalho irá detalhar alguns tipos de interfaces mais comuns encontradas durante a execução destes projetos, as soluções desenvolvidas e a justificativa atrás de cada uma destas escolhas com base nos anos de experiência em implantação de projetos contendo interfaces entre sistemas pela Schneider-Electric.

Palavras-chave: Interface; ERP; Análise.

DEMYSTIFYING INTERFACES BETWEEN AUTOMATION AND CORPORATE SYSTEMS

Abstract

In all MES projects developed by Schneider Electric, a point that always stands out and causes chills in both clients and implementers are the interfaces with enterprise systems. This paper will detail some of the most common types of interfaces encountered during the implementation of these projects by Schneider-Electric, the developed solutions and the rationale behind each of these choices based on years of experience in deployment of projects containing interfaces between systems.

Key words: Interface; ERP; Analysis.

¹ Contribuição técnica ao 68º Congresso Anual da ABM - Internacional, 30 de julho a 2 de agosto de 2013, Belo Horizonte, MG, Brasil.

² Tecnólogo em Processamento de Dados. Coordenador de Projetos a Clientes. Schneider-Electric, Belo Horizonte, Minas Gerais, Brasil.

³ Sistemas de Informação. Arquiteto de Soluções, Schneider-Electric, Belo Horizonte, São Paulo, Brasil.

⁴ Engenharia Controle e Automação. Engenheiro de Aplicação. Schneider-Electric, Belo Horizonte, Minas Gerais, Brasil.

⁵ Engenheiro eletrônico. Coordenador de Serviços MES. Schneider-Electric, Belo Horizonte, Minas Gerais, Brasil.

⁶ Engenharia Controle e Automação. Gerente Comercial, Schneider-Electric, Belo Horizonte, Minas Gerais, Brasil.

1 INTRODUÇÃO

Neste trabalho procuramos mostrar a experiência da Schneider-Electric na implementação de projetos que possuíam algum tipo de interface entre sistemas. Iremos listar os tipos de interface que mais encontramos nestes anos e realizar uma análise qualitativa de sua implementação nos diversos clientes.

Por questões de confidencialidade não iremos mencionar cada um dos clientes e quais interfaces foram utilizadas em cada um, mas sim um resumo e um quadro comparativo da utilização destas tecnologias de interface nestes clientes.

2 MATERIAIS E MÉTODOS

2.1 Materiais

Abaixo serão apresentados, de maneira geral, os materiais que foram utilizados durante os projetos. Posteriormente, durante a discussão, serão melhor detalhados quais materiais ou ferramentas foram utilizadas em cada um dos pontos. Assim sendo, seguem os materiais:

- Documentação – toda a documentação dos projetos foi elaborada utilizando os softwares Microsoft Word em sua versão 2003, 2007 e 2010. Para as interfaces a documentação geralmente consiste em uma ou mais especificações técnicas e funcionais descrevendo como as interfaces serão construídas e suas regras de negócio.
- Ferramentas de desenvolvimento – para os casos que foi necessário desenvolver interfaces específicas, foram utilizadas as ferramentas Visual Studio 2003 e 2008 e também o SQL Development Studio para as interfaces desenvolvidas dentro das ferramentas de banco de dados.
- Ferramenta de MES – todos os projetos utilizaram a ferramenta de MES da Schneider denominada Ampla. Esta ferramenta já traz nativamente várias formas de integração com outros sistemas como OPC, Web Services e Stored Procedures para SQL Server e Oracle.
- *Middlewares* – *middlewares* são sistemas responsáveis por gerenciar a integração entre outros sistemas, controlando as mensagens trocadas entre estes sistemas. Estes softwares possuem várias interfaces nativas e possibilitam a customização das existentes. No caso dos projetos descritos neste trabalho foram feitas integrações com o BizTalk e com o SAP MII, ambos para integrar o MES com o ERP da SAP.

2.2 Métodos

Abaixo serão detalhados os tipos de interfaces utilizadas pela Schneider durante os seus projetos de MES, bem como sua utilização, pontos fortes e pontos fracos.

2.2.1 Interface via arquivo texto

A troca de informações através de arquivos texto é uma das formas mais antigas de comunicação entre sistemas.

Esta comunicação é mais comumente utilizada para troca de dados em batelada.

2.2.1.1 Características principais

Este tipo de comunicação é relativamente fácil de ser implementada e controlada por estar presente em praticamente todas as linguagens de programação nativamente, além de, certamente, estar presente em todos os sistemas operacionais.

De maneira geral a comunicação de arquivo texto é muito utilizada para dados em batelada (sumários, fechamentos etc.) ou para informações de cadastro.

Como principais pontos positivos podemos citar:

- Facilidade de rastreabilidade, visto que normalmente é possível abrir o arquivo texto e visualizar as informações que estão sendo trafegadas
- Facilidade de implementação
- Facilidade de transporte entre redes, com a utilização de mapeamento de diretórios ou mesmo serviços como FTP e HTTP.
- Multiplataforma

Já como pontos negativos, podemos citar:

- Maior dificuldade em implementar interfaces em tempo real devido à arquitetura envolvida
- Maior facilidade para gerar inconsistência dos dados devido à possibilidade de edição manual dos arquivos
- Necessidade de troca de arquivos muito grandes quando o volume de dados é grande gerando muito consumo de rede
- Necessidade de monitoramento das pastas que armazenam estes arquivos para evitar perdas de dados por impossibilidade de geração dos arquivos.

2.2.1.2 Variações da interface de arquivos texto

2.2.1.2.1 Arquivos CSV

Arquivos CSV ou Comma Separated Values ⁽¹⁾ são arquivos texto onde as informações são separadas em registros que são representados por linhas no arquivo e dados que são representados por textos (ou números) separados por vírgulas, sendo esta a origem do nome.

Este tipo de arquivo texto é interpretado pela maioria das planilhas eletrônicas de mercado e também por sistemas de gerenciamento de banco de dados (bancos relacionais), sendo possível tanto exportar quanto importar dados neste formato.

Apesar de não haver um padrão escrito para definir este tipo de arquivo, a RFC 4180 descreve um formato comum e estabelece text/csv como um MIME (Multipurpose Internet Mail Extensions) registrado na IANA (Internet Assigned Numbers Authority).

Como pontos positivos, além dos vistos na interface de arquivo texto temos:

- Sua facilidade de importação em bancos de dados relacionais;
- Importação nativa em muitas ferramentas MES de mercado; e
- Facilidade de visualização dos dados dentro do arquivo.

Seu principal ponto negativo, além dos vistos na interface de arquivo texto é o fato de ser dependente de linguagem, pois, em línguas como o português a vírgula define o separador decimal, o que gera erro na interpretação de números decimais presentes nos arquivos. Para estes casos é necessário trocar o carácter delimitador (normalmente por ponto e vírgula) ou trocar o separador decimal por ponto (formato americano). A necessidade destas alterações acaba aumentando a complexidade de implementação desta interface.

Outra abordagem que traz um resultado mais satisfatório é delimitar os dados utilizando uma série específica de caracteres especiais, o que torna praticamente

impossível a existência de erros, um exemplo seria utilizar a o conjunto de caracteres “!@#\$\$%” como separador ao invés da vírgula ou ponto e vírgula. É necessário, porém verificar se a ferramenta em que os dados serão importados possibilita customização do caracter separador.

2.2.1.2.2 Arquivos XML

XML ou eXtensible Markup Language ⁽²⁾ é um formato definido pela W3C (World Wide Web Consortium) para criação de documentos contendo dados organizados de forma hierárquica.

Sua principal característica é, além de possibilitar a visualização de dados de forma hierárquica, separar o conteúdo da formatação.

Como principais pontos positivos temos:

- é um formato muito utilizado pelo mercado;
- simplifica a legibilidade tanto para humanos quanto para o computador;
- separa o conteúdo da formatação ;
- interpretação nativa pela maioria dos sistemas mes, bancos de dados e *middlewares* presentes no mercado.

Já como pontos negativos temos:

- é um tipo de arquivo com alta verbosidade, ou seja, é necessário ter muito mais texto dentro do arquivo para transferir a mesma informação, se comparado com um arquivo CSV;
- devido à sua alta verbosidade normalmente gera arquivos muito grandes quando se está trafegando muitos dados; e
- menor velocidade de interpretação, visto que normalmente é necessário mapear toda ou boa parte da estrutura do arquivo antes de iniciar a sua utilização.

2.2.2 Interface via COM/DCOM

COM ou Component Object Model ⁽³⁾ é uma tecnologia proprietária da Microsoft que teve sua primeira versão lançada em 1993 e que é utilizada para a criação de componentes de software que podem interagir com outros componentes de software também criados utilizando COM. DCOM ou Distributed COM é a versão para redes da tecnologia COM também criada pela Microsoft.

Já na plataforma .Net, o COM/DCOM foi substituído pelo .Net Remoting ⁽⁴⁾ que é empacotado do WCF (Windows Communication Foundation), porém, para este trabalho, iremos considerar todas as ferramentas listadas abaixo como implementações de COM/DCOM por se tratarem de tecnologias similares.

De maneira geral, a comunicação COM/DCOM é feita através de métodos que são encapsuladas em DLLs (bibliotecas de aplicação) ou no próprio arquivo executável da aplicação e que são disponibilizadas através de um registro na máquina servidora. Um cliente então implementa as interfaces descritas nestas DLLs e posteriormente consome estes métodos e objetos conectando-se localmente (COM) ou remotamente (DCOM) e executando estes métodos disponíveis.

2.2.2.1 Características principais

O COM/DCOM é uma tecnologia amplamente utilizada pelo mercado principalmente porque foi uma das primeiras implementações que tornou possível sistemas distribuídos e multi-camadas. Além de possibilitar comunicação inter sistemas, ela também é bastante utilizada para comunicação intra sistemas, que é o caso de comunicação de sistemas multicamadas.

Como pontos fortes podemos citar:

- possibilidade de implementar interfaces com um baixo acoplamento entre os sistemas;
- desempenho satisfatório, possibilitando a construção de interfaces de tempo real; e
- mecanismo de segurança integrado que possibilita limitar o acesso às informações.

Já como pontos fracos podemos citar:

- limitação de plataformas onde a pesar de a maior parte dos sistemas de automação atualmente utilizarem Windows, ainda são encontrados sistemas legados que são baseados em outros sistemas operacionais, geralmente baseados na arquitetura Unix. Para estes casos, apesar de existirem adaptadores, estes não são comuns e normalmente não são tecnologias dominadas pelas equipes de automação. COM/DCOM também não é implementado na maioria dos sistemas operacionais móveis; e
- problemas de acesso entre redes onde em praticamente todos os clientes em que a Schneider implementou sistemas MES, há uma separação (lógica ou física) entre as redes de TA e TI. Como o MES é um sistema que está posicionado justamente no meio destas redes, é um sistema muito afetado por estas limitações. Para resolver ou contornar estes problemas normalmente são utilizadas ferramentas de tunelamento, que aumentam o custo da solução e também aumentam a complexidade da arquitetura adotada.

Um ponto importante a se salientar é que apesar de possibilitar um baixo acoplamento entre sistemas, é necessário que o cliente conheça e implemente os objetos utilizados pela interface COM.

2.2.2.2 Interfaces que utilizam a tecnologia COM/DCOM

2.2.2.2.1 APIs nativas

Vários sistemas de mercado disponibilizam APIs nativas. Estas APIs trazem métodos e objetos específicos que possibilitam à uma aplicação externa acessar ou mesmo alterar as informações daquele sistema.

Em alguns casos as APIs nativas não são implementadas através de COM/DCOM mas sim através de sockets de comunicação. Estes sockets são portas específicas abertas entre cliente e servidor que possibilitam a troca de informações. Esta implementação costuma trazer um desempenho melhor, porém, aumenta o acoplamento pois você precisa anexar ao seu sistema uma biblioteca específica e, em caso de troca de versão, é necessário recompilar o sistema cliente para a nova versão de biblioteca. Uma outra vantagem é que por utilizar portas específicas, seu roteamento entre redes é simplificado.

Esta abordagem é comum em sistemas PIMS e Supervisórios.

2.2.2.2.2 OPC DA

OPC ou Open Productivity and Connectivity⁽⁵⁾ é um conjunto de padrões criados e mantidos pela OPC Foundation que tem como principal objetivo garantir interoperabilidade dentro da automação. Esta fundação cria e mantém especificações abertas que padronizam a aquisição de dados de processo, eventos, alarmes, dados históricos e dados de batelada entre diferentes sistemas

corporativos de várias empresas desenvolvedoras, além de dispositivos de produção.

O OPC DA ou OPC Data Access é um padrão de comunicação baseado em OLE COM/DCOM que define uma série de objetos, interfaces e métodos para utilização em controle de processo e aplicações de automação para facilitar a interoperabilidade e possibilitar a comunicação em tempo real entre aplicações.

O objetivo principal deste padrão foi possibilitar que as empresas fornecedoras de informações (supervisórios, PLCs, etc) pudessem criar servidores de informações que posteriormente eram consumidos por clientes, também padronizados, pelos sistemas da empresas como PIMS, MES, etc. Desta forma passou a ser possível aos clientes finais escolher entre ferramentas pelas suas características e não porque a ferramenta A possuía driver para o PLC X.

A OPC Foundation escolheu utilizar o COM/DCOM justamente por se tratar de um padrão de mercado. As versões mais recentes do padrão já são implementadas utilizando a tecnologia .Net Remoting.

Além dos pontos positivos e negativos da tecnologia COM/DCOM, o padrão OPC DA tem como principal ponto positivo a padronização da comunicação entre as aplicações. Já sua limitação está no fato de tratar apenas dados em tempo real, ou seja, não é possível buscar dados históricos ou outros tipos de informações, tendo seu uso limitado à informações de processo (leitura de equipamentos, status de equipamentos, etc.).

2.2.2.2.3 OPC HDA

O OPC HDA ou OPC Historical Data Access é bastante semelhante à versão DA do OPC, diferindo principalmente no tipo de dado trafegado, já que o OPC HDA define o padrão de objetos, interfaces e métodos para troca de informações históricas.

Os pontos positivos e negativos do OPC HDA são os mesmos do OPC da, excetuando o fato de ser possível buscar dados históricos.

Como uma limitação importante, não existe um método que traga somente as informações alteradas naquele período, sendo que o cliente deverá fazer este tratamento, caso seja necessário.

2.2.3 Interface via tabelas de bancos de dados

Interfaces via tabelas de banco de dados são as interfaces feitas através de acesso direto à uma ou mais tabelas de uma aplicação por outra aplicação. Este acesso pode ser feito de diversas maneiras, conforme descrito abaixo, e muitas vezes é controlado pelo próprio SGBD (Sistema de Gerenciamento de Banco de Dados).

Sua principal utilização é quando a aplicação A precisa dos dados que estão na aplicação B. Os desenvolvedores da aplicação B então disponibilizam os dados desta aplicação para os desenvolvedores da aplicação A que então usam estes dados.

O grau de acoplamento deste tipo de interface vai variar de acordo com sua implementação, podendo variar de altamente acoplado para um acoplamento leve.

Como pontos fortes podemos citar:

- facilidade de implementação devido à familiaridade dos desenvolvedores com este tipo de interface;
- desempenho satisfatório na maior parte das implementações devido às otimizações nativas das ferramentas de banco de dados mais atuais;
- normalmente é implementado de maneira nativa pela maioria das aplicações MES e *middlewares*.

Já como pontos fracos podemos citar:

- alto acoplamento entre as aplicações, visto que mais de uma aplicação precisa acessar a mesma tabela para ler e, às vezes, alterar informações; e
- dificuldade de rastreabilidade pois mais de uma aplicação está acessando os mesmos dados.

2.2.3.1 Formas de implementar interfaces via banco

2.2.3.1.1 Acesso direto às tabelas de uma aplicação pela outra

Este tipo de acesso se dá quando a aplicação A acessa diretamente os dados da aplicação B. Este acesso normalmente é feito apenas como leitura e este acesso é sempre controlado dentro do próprio SGBD.

Esta abordagem gera um alto acoplamento entre os sistemas, visto que qualquer alteração na tabela do sistema A afetará diretamente o sistema B.

2.2.3.1.2 Acesso via *views* por uma aplicação

Já neste tipo de acesso o sistema A disponibiliza uma *View* (ou visualização) dos seus dados em um formato específico, o sistema B então acessa esta *View* (que normalmente é somente leitura) e busca os dados que são necessários. Esta *view* é vista como uma tabela separada que pode ser acessada normalmente como qualquer tabela do banco.

Essa abordagem gera um acoplamento menor, visto que as tabelas da aplicação A podem ser alteradas sem necessariamente alterar a *View* que é utilizada pela aplicação B.

Normalmente *views* são utilizadas quando se quer limitar o acesso aos dados de uma aplicação. Muitos SGBDs também possuem ferramentas que aceleram o acesso aos dados de uma determinada *view*, podendo com isso aumentar a velocidade de uma aplicação.

2.2.3.1.3 Acesso via *Stored Procedures* por uma aplicação

É um sistema de acesso similar ao acesso via *Views*, porém, ao invés de se ter uma *view* criada como um objeto do SGBD, estas *procedures* contêm *SELECTs* customizados que são armazenados como procedimentos.

Possui os mesmos pontos positivos e negativos do acesso via *View*, sua diferença básica é a maneira como as aplicações fazem a chamada a estas *procedures* pois, ao invés de selecionar as informações de uma tabela, uma *procedure* é executada e esta retorna um conjunto de valores.

2.2.3.1.4 Cópia dos dados entre tabelas de banco sem uso de aplicações externas

Conhecida também como replicação de dados entre tabelas este tipo de abordagem utiliza de *Jobs* (tarefas) internas do SGBD para copiar dados de uma tabela de uma aplicação para outra.

Seu acoplamento é extremamente alto visto que as tabelas devem ser, de preferência, idênticas para facilitar esta cópia.

Um ponto importante a se levar em conta é que a maioria das equipes de automação não contam com um profissional DBA (*DataBase Analyst*) para executar e manter este tipo de interface, tornando a equipe dependente de serviços externos.

Esta abordagem também dificulta a manutenção de aplicações pois passa a existir uma regra de negócio fora do domínio da aplicação, o que normalmente é considerado um problema de arquitetura.

2.2.3.1.5 Utilização de tabelas de interface para comunicação entre aplicações

Tabelas de interface são tabelas criadas com o único propósito de troca de informações entre sistemas.

Sua implementação mais comum consiste em uma tabela que pode ser vista e editada por duas aplicações, sendo uma das colunas um controle de status, onde as aplicações devem “se comunicar” informando quando uma leu ou escreveu uma informação daquela tabela.

Sua grande desvantagem está no fato de duas aplicações poderem acessar e alterar dados simultaneamente, podendo causar erros de consistência e perda de informações.

Porém, este tipo de abordagem gera um acoplamento menor pois as aplicações podem alterar suas tabelas de forma independente, ficando o acoplamento restrito às tabelas de interface.

2.2.4 Interface via *web services*

Web Services(6) ou Serviços Web são implementações de SOA (*Service Oriented Architecture*) através de protocolos de internet padrão e independentes de arquitetura e linguagem de programação.

Seu principal objetivo é fornecer meios padrão de interoperabilidade entre diferentes *softwares*, sendo executados em diferentes plataformas e/ou *frameworks*. *Web Services* são caracterizados pela grande interoperabilidade e escalabilidade e também pelas suas descrições fáceis de serem processadas pelas máquinas devido à utilização de XML.

Em grandes empresas os *Web Services* são publicados através de barramentos, chamados de barramentos de serviço, simplificando e controlando o acesso à estes serviços, sendo que o principal objetivo é reduzir a duplicidade de informações em uma rede.

Os pontos forte da utilização de *Web Services* são:

- baseado em tecnologias abertas, amplamente divulgadas e padronizadas;
- independente de plataforma, sistema operacional e linguagem de programação;
- facilidade de serem consumidos por diversos sistemas devido a serem descritos utilizando XML e uma linguagem padrão (WSDL);
- possibilita versionamento de interfaces, aumentando o desacoplamento entre sistemas;
- estarem disponíveis nativamente nos principais sistemas PIMS, MES e ERP de mercado, facilitando a interligação entre estes sistemas;
- são escaláveis mais facilmente que outras tecnologias; e
- tem se tornado o padrão mais utilizado de comunicação entre sistemas de TI.

Como pontos negativos podemos destacar:

- diferente do COM/DCOM, os *Web Services* não possuem serviços de segurança nativos, sendo necessário implementar esta segurança diretamente nas aplicações; e
- menor desempenho se comparado com as soluções RFP (como COM/DCOM) devido à troca de informações via texto/XML ao contrário de

arquivos binários, principalmente quando se trata de grande volume de dados.

3 RESULTADOS

Como resultado deste trabalho iremos apresentar uma tabela contendo uma qualitativa das aplicações de cada uma das interfaces em seus respectivos mercados, bem como suas características técnicas.

Para facilitar a visualização iremos utilizar uma escala de estrelas, variando de zero a duas estrelas, onde:

- nenhuma estrela: não se aplica ou não atende sendo necessário desenvolvimento à parte;
- uma estrela: atende de maneira parcial, sendo necessário algum tipo de customização do lado do cliente ou do servidor; e
- duas estrelas: atende de maneira nativa e satisfatória.

Com relação ao mercado, estaremos utilizando o conceito descrito na ISA95 de camadas, considerando a visão de TI da mesma representada na Figura 1.

Topologia conceitual – Visão da TI

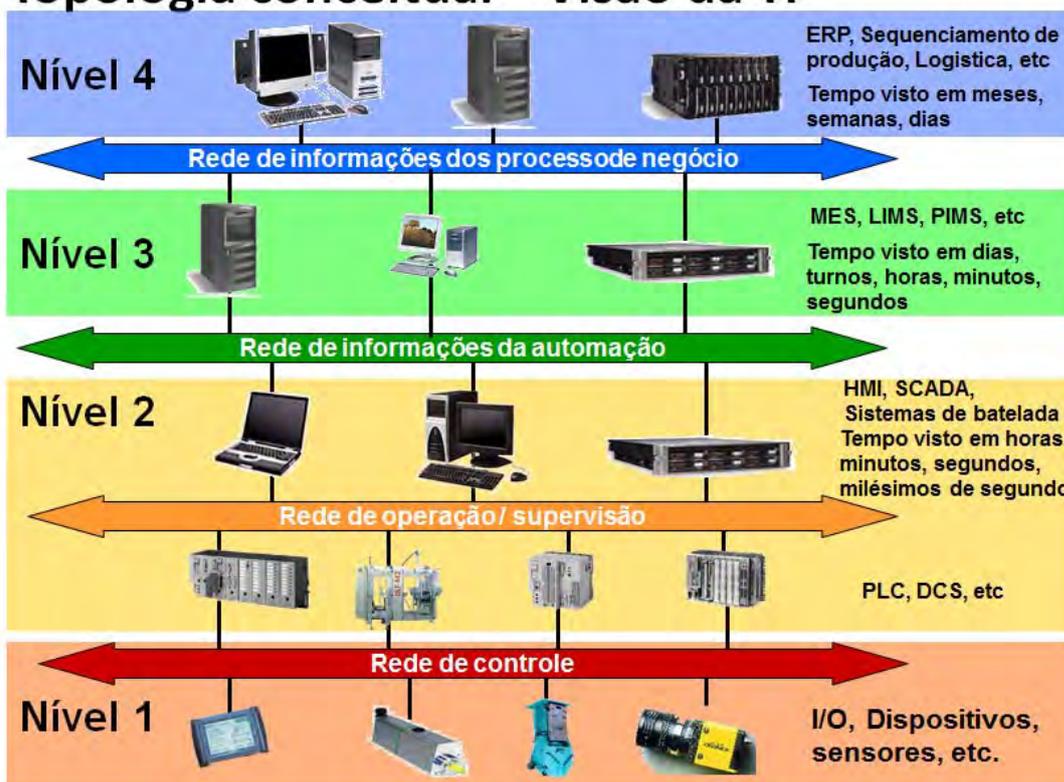


Figura 1. Topologia baseada no conceito de camadas da ISA 95.⁽⁷⁾

Foram analisados os seguintes critérios:

- rastreabilidade: possibilidade de visualizar os dados que foram trafegados de forma nativa, sem uso de ferramentas específicas ou desenvolvimento/customização;
- facilidade de implementação: se é baseada em algum padrão de mercado ou norma facilmente encontrada, podendo ser grátis ou paga;
- multiplataforma: se pode ser utilizado de forma nativa em várias plataformas;

- segurança: se implementa algum tipo de segurança de forma nativa, sem necessidade de customização;
- aceitação TA: se é aceita pelas ferramentas e sistemas de TA de forma nativa, sem necessidade de customização;
- aceitação TI: se é aceita pelas ferramentas de TI, principalmente pelos *Middleware*s, sistemas MES e ERPs;
- velocidade: capacidade de implementar interfaces em tempo real baseada na verbosidade da interface e nas características de implementação da tecnologia; e
- dados customizados: possibilidade de se trafegar dados customizados ou específicos da aplicação, inclusive com hierarquia das informações.

Tabela 1. Resultados da análise qualitativa das interfaces

Interface	Rastreabilidade	Implementação	Multiplicação	Segurança	Aceitação TA	Aceitação TI	Velocidade	Dados Customizados
Texto CSV	★★★	★★★	★★★	★	★★★	★★★	★	★
Texto XML	★★★	★★★	★★★	★		★★★		★★★
API Nativa		★			★★★		★★★	★★★
OPC DA		★★★	★	★	★★★	★	★★★	
OPC HDA	★	★★★	★	★	★	★		
Tabelas SGBD	★	★★★	★★★	★★★	★★★	★★★	★	★
Web Services	★	★★★	★★★	★	★	★★★	★	★★★

4 DISCUSSÃO

Com relação aos resultados apresentados, gostaríamos de salientar dois pontos importantes:

- este trabalho não visa e não deve ser utilizado para indicar se uma determinada interface foi implementada ou utilizada de forma correta, visto que entendemos que o mercado de tecnologia é muito dinâmico e, além disso, a utilização de um ou outro tipo de interface varia de acordo com vários outros fatores não analisados; e
- a questão da Velocidade de cada interface é um assunto polêmico e deve ser analisado caso a caso, a representação do resultado visa mostrar o que foi encontrado pela Schneider-Electric nas suas implementações de interface com sistemas de TA e TI, sendo que as mesmas podem variar de acordo com as tecnologias de transporte (rede), hardwares e até da própria implementação da interface.

4.1 Rastreabilidade

Quando falamos de rastreabilidade, a importância desta característica está no fato de, na maior parte das implementações de interfaces, estas são implementadas por equipes diferentes, deste modo a rastreabilidade se torna importante pois possibilita que as duas equipes saibam exatamente o que está sendo trafegado, evitando problemas de comunicação e facilitando o reteste.

No caso dos arquivos texto, caso haja algum problema basta reprocessar o arquivo pois ele normalmente não é apagado mas sim movido ou marcado como já lido.

Já para as interfaces via *WebServices*, SGBD e OPC HDA, apesar de não ser algo visual, ou seja, não é possível que o ser humano visualize os dados que foram trafegados, é possível através de ferramentas próprias visualizar os dados, se estes são os mesmos desde o momento do envio e, caso sejam, é possível reprocessar e reenviar as informações.

Já no caso de APIs Nativas e OPC DA, eles normalmente tratam de informações instantâneas, ou seja, caso a informação tenha sido alterada não é possível reprocessá-la.

4.2 Implementação

Para este caso consideramos que todos os tipos de interface atendem ao quesito informado, exceto as APIs nativas pois normalmente elas são específicas de um fornecedor e, caso este fornecedor não forneça a documentação da API, é muito difícil encontrar informações sobre os métodos e objetos disponíveis.

4.3 Multiplataforma

As APIs nativas normalmente são feitas para uma plataforma específica, ou seja, normalmente não podem ser utilizadas em outros ambientes ou sistemas operacionais caso o fornecedor não tenha uma versão para esta plataforma.

No caso do OPC DA e HDA, por eles serem baseados no protocolo COM/DCOM da Microsoft eles ficam limitados à esta plataforma. Embora existam implementações de COM/DCOM para outras plataformas, estas não são suportadas pelo dono do padrão que é a Microsoft, ficando o usuário dependente de um desenvolvedor à parte.

É importante salientar que existe um novo padrão proposto pela OPC Foundation para implementação de um OPC multiplataforma, porém, esta abordagem ainda não é muito comum no mercado.

4.4 Segurança

No caso das interfaces via arquivo texto, apesar de estas não apresentarem uma segurança propriamente dita, podendo qualquer usuário alterar seus dados, elas estão apoiadas na segurança do sistema de arquivos do sistema operacional onde são utilizadas. Por isso consideramos que há alguma forma de implementação de segurança, embora esta não seja uma forma de segurança recomendada. Um ponto importante a ser mencionado é que em muitas redes de automação ou controle não há um domínio ou um controle de usuário mais efetivo, diminuindo esta segurança.

As APIs nativas que trabalham via Socket devem implementar a segurança e por isso não receberiam estrelas, porém, as implementadas via COM/DCOM, como também as interfaces via OPC DA e HDA utilizam a segurança nativa do Windows para controle de usuários. Além disso, as informações não ficam disponíveis para alteração, além dos sistemas que os implementam também apresentarem um controle de segurança próprio, por isso foi considerado que elas atendem parcialmente.

As interfaces via Web Service apesar de poderem ter um nível de segurança de acesso baseado no servidor Web, normalmente não os implementam, sendo que o mais comum é que a segurança fique a cargo da aplicação que contém o Web Service.

Os SGBD comumente apresentam um nível de segurança mais elevado que os demais, tendo políticas de senha e segurança próprias e independentes da aplicação. Deste modo consideramos que o mesmo tenha duas estrelas nesta análise.

4.5 Aceitação TA

Este tópico foi baseado na experiência da Schneider-Electric em implementar interfaces, portanto diferentes empresas ou pessoas podem classificar as interfaces de maneira diferente.

Uma característica importante das equipes de TA é que as equipes normalmente primam por estabilidade e por isso elas tendem a procurar soluções mais estáveis e já testadas por um bom tempo. Deste modo é compreensível que as novas tecnologias ou padrões demorem um tempo para se estabelecerem e adentrarem neste mercado.

Deste modo consideramos que as interfaces mais aceitas pela TA por serem as mais comuns de serem encontradas são:

- supervisórios e PLC: APIs nativas ou OPC DA;
- PIMS: Arquivo CSV, OPC DA, OPC HDA e APIs Nativas;
- LIMS: Banco de dados; e
- outros: banco de dados, arquivo texto ou padrões próprios (ex.: área de energia).

Vemos também algumas iniciativas em se implementar interfaces via Web Services, porém estas ainda não estão consolidadas neste mercado.

4.6 Aceitação TI

Este tópico foi baseado na experiência da Schneider-Electric em implementar interfaces, portanto diferentes empresas ou pessoas podem classificar as interfaces de maneira diferente.

Nos casos analisados pela Schneider-Electric, a interface com a TI envolve os sistemas MES e/ou PIMS com o ERP/Middlewares ou então sistemas de nível 2 diretamente com o ERP.

Para estes casos vemos uma preferência para utilização de interfaces via Web Services, sendo que esta vem se tornando um padrão de mercado.

Porém, ainda encontramos muitas implementações via arquivo texto (CSV) e via tabelas de SGBDs, sendo estes principalmente com relação às interfaces diretas com os sistemas de nível 2 e/ou sistemas legados ondem não há suporte para Web Services.

Em alguns casos temos também equipes de TI mais conservadores, com características semelhantes às equipes de automação, nestes casos as interfaces utilizadas são normalmente via tabelas do SGBD.

4.7 Velocidade

Este é um tópico muito polêmico pois a velocidade ou desempenho de uma interface varia muito de acordo com o ambiente que esta interface está implementada.

Porém, de maneira geral, podemos separar as interfaces entre Tempo Real e Batelada.

As interfaces de tempo real, que precisam de informações em segundos ou milissegundos, normalmente são implementadas utilizando APIs nativas. Para os casos onde segundos e minutos são aceitos, a implementação via OPC DA é mais comum.

Todas as demais interfaces são normalmente utilizadas para informações de batelada, ou seja, não é necessária uma troca instantânea de um volume muito grande de informações. No caso dos *Web Services* estes podem até ser utilizados para interfaces de tempo real, porém, seu desempenho vai variar significativamente de acordo com o volume de dados a ser trafegado.

Novamente é importante salientar que existem implementações específicas destas interfaces que possibilitam uma velocidade maior, porém, estas não são as mais comuns e por isso não foram consideradas neste trabalho.

4.8 Dados Customizados

Quando é necessário trafegar dados customizados, as interfaces que possuem um padrão de dados como o OPC DA e HDA não são utilizáveis, visto que estas são específicas para dados temporais.

No caso dos arquivos CSV e tabelas de SGBDs, normalmente eles são customizáveis, porém, sem a possibilidade nativa de se implementar valores hierárquicos. Com isso, em alguns casos, é necessário repetir informações nas diversas linhas que estão sendo trafegadas.

Um exemplo seria um dado de receita, onde se deve trafegar o material e os seus ingredientes. Em arquivos CSV e tabelas SGBD o código do material que se está produzindo deve ser repetido em cada uma das linhas dos ingredientes, para que se possa fazer um relacionamento.

Já no caso de arquivos XML ou *Web Services* é possível implementar uma hierarquia que pode ser navegada, deste modo a estrutura fica mais visível e menos dados são trafegados.

Para o caso das APIs nativas, elas podem trafegar objetos customizados que podem ou não implementar informações hierárquicas.

5 CONCLUSÃO

Como conclusão deste trabalho temos uma análise qualitativa de como e onde as interfaces são comumente utilizadas no mundo de TA e TI.

As informações aqui encontradas podem ser utilizadas como base para um trabalho mais detalhado, envolvendo testes quantitativos de implementação de cada uma das interfaces em cenários específicos.

De forma geral esperamos poder facilitar e ajudar a desmistificar este assunto tão importante que é a interface entre sistemas, mostrando uma visão clara e simples de qual tipo de interface mais se adequa à uma ou outra implementação.

BIBLIOGRAFIA

- 1 Shafranovich, Y. - *Common Format and MIME Type For Comma Separated Values* - Disponível em <http://tools.ietf.org/html/rfc4180>, última visualização em 11/03/2013
- 2 Bray, T.; Paoli, J. - *Extensible Markup Language (XML) 1.0 (Fifth Edition)*. Sperberg-McQueen, C. M.; Maler, E.; Yergeau, F.. Disponível em: <http://www.w3.org/TR/REC-xml/>, última visualização em 11/03/2013
- 3 *COM and DCOM OS Design Development* - <http://msdn.microsoft.com/pt-br/library/aa910773.aspx> - 29/08/2008 - visualizado em 11/03/2013
- 4 Rogerson. D.; *Inside Com - Microsoft Programming Series*, Microsoft GmbH, 1997
- 5 *About OPC* - OPC Foundation - Disponível em http://www.opcfoundation.org/Default.aspx/01_about/01_whatIs.asp?MID=AboutOPC, última visualização em 11/03/2013
- 6 R. Chinnici, J-J. Moreau, A. Ryman, S. Weerawarana, - *Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language*, Editors. World Wide Web Consortium, 23 May 2007. Disponível em: <http://www.w3.org/TR/2007/PR-wsdl20-20070523>. Último acesso em 31/03/2013.
- 7 Brand, D; *The IT Implications of ISA 95 and ISA 99*. BR&L Consulting, Inc – 2005 – Disponível em: <http://www.brlconsulting.com/Files/The%20IT%20Implications%20S95%20and%20S99.pdf>, última visualização em 26/04/2013