



**VI CONGRESSO NACIONAL DE ENGENHARIA MECÂNICA**  
**VI NATIONAL CONGRESS OF MECHANICAL ENGINEERING**  
**18 a 21 de agosto de 2010 – Campina Grande – Paraíba - Brasil**  
*August 18 – 21, 2010 – Campina Grande – Paraíba – Brazil*

## **DESENVOLVIMENTO DE UM SISTEMA DE MONITORAMENTO PARA MÁQUINAS-FERRAMENTAS UTILIZANDO-SE O PADRÃO MTCONNECT®**

**Eraldo Jannone da Silva, eraldojs@sc.usp.br<sup>1</sup>**  
**Leandro Coelho Rondon, rondonusp@gmail.com<sup>2</sup>**  
**Mariana Vilela Martins, mariana.mk07@gmail.com<sup>2</sup>**  
**Thiago Valle França, thiagovf@sc.usp.br<sup>1</sup>**  
**José Eduardo Cervelin, cervelin@usp.br<sup>1</sup>**

<sup>1</sup>Universidade de São Paulo - USP - Escola de Engenharia de São Carlos- EESC; Departamento de Engenharia de Produção – SEP. Av. Trabalhador São-carlense, 400, CEP: 13566-590, São Carlos – SP, Brasil.

<sup>2</sup>Universidade de São Paulo - USP - Instituto de Ciências Matemáticas e de Computação – ICMC. Av. Trabalhador São-carlense, 400, CEP: 13560-570, São Carlos – SP, Brasil.

**Resumo.** *Com o desenvolvimento das máquinas-ferramentas CNCs, a coleta de informações do processo de forma detalhada e em tempo real tem se tornado um diferencial competitivo para aqueles que adotam esta estratégia. No entanto, a aquisição de dados de forma integrada em linhas de produção com elevados números de máquinas-ferramentas com CNCs distintos ainda é um grande desafio, já que a individualização de sistemas de monitoramento/supervisão torna-se necessário, em virtude do caráter distinto e proprietário dos protocolos de comunicação destes controles. O formato MTConnect® foi criado com o intuito de se tornar um padrão entre fabricantes de máquinas CNC, assim como o Bluetooth ou o USB em suas respectivas áreas. Com a adoção de um formato único de entrada e saída de dados, a implementação de sistemas de controle e análise desses dados seria facilitada. No entanto, as máquinas CNC atuais não oferecem suporte a dados no padrão MTConnect®, sendo necessário o desenvolvimento de um aplicativo adaptador, o qual faz a "tradução" dos dados para o padrão proposto. Ele pode ser instalado em um computador que se comunique com o CNC ou mesmo no próprio CNC (sendo de arquitetura aberta). Além disso, o formato utiliza o protocolo HTTP para o envio de dados, possibilitando o monitoramento remoto de várias linhas de produção simultaneamente. O objetivo deste trabalho foi o desenvolvimento de um sistema de monitoramento utilizando-se o protocolo MTConnect® para a coleta on-line dos principais parâmetros de processo num centro de usinagem CNC, tais como: posições atuais dos eixos, rotação da ferramenta, status do CNC, dentre outros. Os elementos da arquitetura foram desenvolvidos e testados, incluindo um adaptador para CNC Fanuc. Como resultado foi possível a coleta remota via protocolo HTTP dos principais parâmetros-máquina.*

**Palavras-chave:** *usinagem, monitoramento, CNC*

### **1. INTRODUÇÃO**

São vários os fatores, tais como custos, qualidade, flexibilidade e tempo de resposta, que são diferenciais de competitividade nas indústrias. No ramo metal-mecânico, esses fatores estão intimamente ligados ao sistema de produção, especificamente ao chão-de-fábrica e à sua gestão. Se a gestão da produção não estiver suficientemente preparada com informações precisas, detalhadas e em tempo real, esses fatores de competitividade podem ser afetados negativamente (Ferraz Júnior, 2007).

De acordo com Rockwell (2004), historicamente, o chão-de-fábrica tem sido visto de forma isolada. Grandes cuidados são tomados para isolar as plantas de fabricação dos riscos de escassez de fornecedores ou flutuações na demanda do consumo. As fábricas operam como unidades autônomas, com seus próprios objetivos, como diminuir o custo de produção por unidade ou armazenar estoques. No entanto, essa visão ignora um fator essencial no mercado pós-Internet: a resposta às necessidades do consumidor. O acesso à informação está se tornando mais importante que nunca, pela necessidade de entender a verdadeira capacidade das plantas de produção e assim torná-las mais eficientes. Neste âmbito, à medida que cresce a demanda por processos de manufatura enxutos, ágeis e em conformidade com os requisitos ambientais, aumenta-se a necessidade da automação dos sistemas de manufatura (Difrank, 2008). Estes têm se difundindo na indústria moderna, integrados aos sistemas de gerenciamento da produção (MES- *Manufacturing Execution Systems*) e aos sistemas de planejamento das atividades corporativas (ERP – *Enterprise Resource Systems*),

atingindo assim todos os níveis da cadeia de automação (Sauter, 2005). As informações coletadas no chão-de-fábrica podem possuir diferentes resoluções temporais (dias, horas, minutos, segundos ou frações de segundos), de acordo com o propósito principal de automação: supervisão da produção, monitoramento da produção e/ou monitoramento da ferramenta, máquina ou processo. Assim, a indústria utiliza diferentes arquiteturas para atender as necessidades em cada nível da cadeia de automação. A disseminação de CNCs com IHM (interface homem-máquina) aberta permitiu a implementação e a integração de sistemas de monitoramento utilizando-se todos os recursos já disponíveis numa máquina-ferramenta, tais como: sensores, rede de comunicação, condicionadores/amplificadores de sinais, base de dados, softwares aplicativos, etc. (Oliveira et al., 2008).

No entanto, a aquisição de dados de forma integrada em linhas de produção com elevados números de máquinas ainda é um grande desafio. Não há escassez de dados no chão-de-fábrica, pelo contrário, os equipamentos de automação industrial modernos, controlados por *software* avançado e interconectado por redes abertas geram uma grande quantidade de dados sobre o sistema de manufatura. Porém, o uso de máquinas de diferentes fabricantes implica o uso de diferentes padrões de entrada e saída de dados. Essas diferenças podem incluir o uso de diferentes tipos de cabos, aplicativos, linguagens de programação ou mesmo sistemas operacionais, o que torna extremamente trabalhoso projetar e configurar um sistema que possa se comunicar e coletar dados sobre o funcionamento de todas essas máquinas ao mesmo tempo. Esses obstáculos fazem com que os sistemas de monitoramento tenham que ser projetados especialmente para cada planta, gerando maiores custos e diminuindo a eficiência.

O formato MTConnect® foi criado pela MTConnect Institute, com o intuito de se tornar um padrão entre fabricantes de máquinas CNC, assim como o Bluetooth ou o USB em suas respectivas áreas (Warndorf et al., 2007). Com a adoção de um formato único e largamente utilizado de entrada e saída de dados, a implementação de sistemas de controle e análise desses dados seria grandemente facilitada. Além disso, o formato utiliza o protocolo HTTP para o envio de dados, possibilitando o monitoramento remoto de várias linhas de produção simultaneamente.

No entanto, as máquinas CNC atuais não oferecem suporte a dados no padrão MTConnect, precisando de um adaptador para se integrarem a linhas de produção nesse padrão. Esse adaptador consiste em um software desenvolvido especialmente para o CNC em questão, que faz a “tradução” dos dados do formato proprietário da máquina para o padrão MTConnect. Ele pode ser instalado em um computador que se comunique com o CNC ou mesmo no próprio CNC (em caso de CNCs de arquitetura aberta), possibilitando a troca de dados.

Esse projeto visa desenvolver e implementar em uma máquina-ferramenta CNC um adaptador para o formato MTConnect, permitindo que os dados obtidos durante sua operação sejam enviados para um servidor, onde poderão ser analisados sem conflitos relacionados a formatos proprietários.

## 2. O PADRÃO MTCONNECT®

Em um dia típico em uma planta de manufatura, centenas ou milhares de máquinas e sistemas independentes operam em união para garantir que um produto seja manufaturado rapidamente, com qualidade e com custo baixo (Warndorf et al., 2007). Cada uma dessas máquinas e sistemas acumula informações sobre sua operação e, normalmente, não pode comunicar essa informação a outros sistemas. Como resultado, a coordenação, otimização e análise de dados para assegurar que a máquina, fábrica ou sistema está operando em um nível aceitável (eficiência, sincronização, uso de energia, erros de forma) é muito difícil.

Além disso, muitas vezes as máquinas existentes são equipadas com sensores adicionais ou outros equipamentos para melhorar a qualidade ou a taxa de medições originais. Os dados gerados por esses equipamentos na maioria das vezes não são integrados às medições do equipamento original, tornando ainda mais complexa a integração de grandes quantidades de máquinas.

Aqui ocorre o oposto do que vemos na área de computação ou tecnologia da informação. Nelas, uma grande quantidade de dispositivos diferentes se comunica utilizando interfaces padronizadas (como o USB) e protocolos padronizados de comunicação. O desafio é, portanto, como a manufatura pode replicar o sucesso dessa interconectabilidade.

Tentativas foram feitas no passado para suprir essa necessidade, mas nenhuma se consolidou como um meio comum para comunicação. O MTConnect<sup>SM</sup> Institute criou uma solução diferente para o problema. A idéia não é criar hardware ou software especial para conectar máquinas e sistemas, mas projetar um padrão aberto de comunicação que se espelha no sucesso da área de tecnologia da informação. Assim dispositivos, equipamentos e sistemas usariam para a saída de dados um formato padrão que possa ser lido por qualquer outro dispositivo. O padrão, MTConnect, é baseado em XML (*Extensible Markup Language*), que oferece uma representação flexível para a troca de dados semi-estruturados. O padrão é aberto e livre de royalties para assegurar a sua adoção. Espera-se que a interoperabilidade do padrão estimule o desenvolvimento de software e hardware de terceiros, tornando o processo de manufatura mais produtivo.

A utilização do padrão beneficiaria o desenvolvimento de sistemas de monitoramento e supervisão, principalmente quando da necessidade do monitoramento de máquinas-ferramentas que apresentam CNCs de fabricantes distintos, cada qual com seu padrão proprietário de comunicação. Em oposição à necessidade de réplica e individualização do sistema de monitoramento/supervisão para a coleta de dados comuns às máquinas em questão (ex. potência e rotação do eixo, posições absolutas, etc.) a utilização da camada intermediária MTConnect permitirá que um sistema único de monitoramento/supervisão se comunique com todas as máquinas CNCs distintos, conforme apresentado na Figura 1.

## 2.1. Componentes do padrão

Os principais componentes do Padrão MTConnect serão descritos a seguir:

### 2.1.1. Adaptador

Por ser o MTConnect um padrão em desenvolvimento, a grande maioria dos CNCs é incompatível com ele. Para tornar essa compatibilidade possível, é necessário o uso de um adaptador. O adaptador realiza a “tradução” entre o formato proprietário utilizado pelo CNC e o padrão MTConnect. A parte de hardware do adaptador inclui qualquer hardware necessário para adquirir os dados da máquina ou do sensor, como cabos de padrão proprietário, e torná-los acessíveis a um PC. A parte de software do adaptador “traduz” os dados adquiridos do formato proprietário para o padrão MTConnect. Do ponto de vista do ambiente MTConnect, o CNC não-compatível, quando associado ao adaptador, passa a ser compatível.

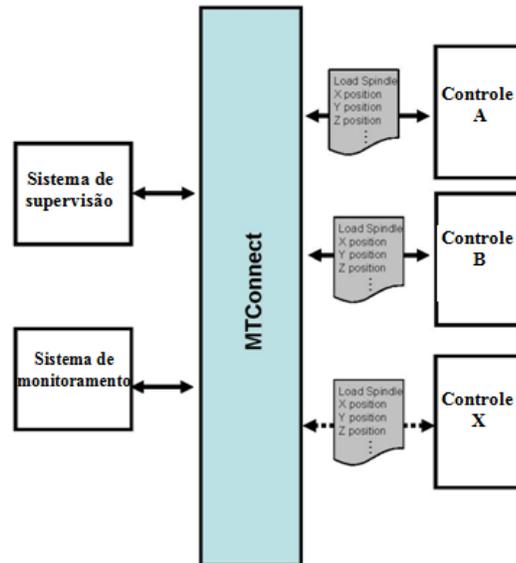


Figura 1 – Cenário de comunicação via padrão MTConnect.

### 2.1.2. Agente

Agente MTConnect é um software que reúne os dados enviados por dispositivos compatíveis com o padrão MTConnect, organizando-os em uma base de dados construída com o padrão XML. Por ser este um padrão largamente adotado, existem várias ferramentas disponíveis em diversas linguagens de programação para sua decodificação e análise. Essas ferramentas facilitam a criação de aplicativos para análises específicas dos dados coletados pelo agente.

O software Agente utilizado nesse projeto é fornecido pelo MTConnect Institute, juntamente com seu código-fonte, que pode ser livremente modificado. Ele foi desenvolvido na linguagem C#, utilizando o Microsoft Visual Studio. Por ser fornecido na forma de código-fonte, o Agente pode tanto ser utilizado como um aplicativo independente quanto como parte da funcionalidade de outros aplicativos.

A comunicação com o agente é feita através do protocolo HTTP, baseado em requisições e respostas, desenvolvido para sistemas distribuídos, colaborativos e multimídias, sendo o principal protocolo de comunicação utilizado pela internet (IETF, 2010). O cliente — navegador ou outro sistema que fará a requisição — solicita um determinado recurso (uma página ou serviço) através do envio de um pacote de informações contendo cabeçalhos a um URI ou, mais especificamente, URL. O servidor recebe estas informações e envia uma resposta, que pode ser um recurso, dados ou um simplesmente outro cabeçalho. Dessa forma, um agente MTConnect funciona como um servidor HTTP, suportando requisições através do método HTTP GET e gerando a resposta em um documento XML.

A comunicação com um adaptador se dá quando esse faz uma requisição HTTP ao servidor (agente) enviando os dados lidos do CNC no corpo dessa requisição, e o agente responde com um *XML-Acknowledgement* (XML-ACK), um documento XML contendo o horário em que o *HTTP Request* foi recebido, e quaisquer erros que possam ter ocorrido durante o armazenamento dos dados. Por outro lado, as aplicações clientes (aquelas que irão utilizar os dados coletados) também fazem requisições HTTP ao agente, podendo ter como resposta um documento XML no formato MTConnect, contendo as configurações do agente, os dados mais atuais ou as amostras dos dados.

### 2.1.3. Clientes

Clientes são softwares que se conectam ao Agente para requisitar informações com algum propósito. Todo o padrão MTConnect é desenvolvido como uma estrutura para que Clientes possam ser desenvolvidos sem preocupação com

diferenças de formato entre os dados coletados em diferentes máquinas. A análise dos dados feita pelo Cliente não é relevante para esse projeto, sendo importante somente que todas as informações no CNC possam completar o trajeto “CNC– adaptador–agente–cliente” de forma eficiente e sem perdas.

## 2.2. Devices.xml: A definição do padrão MTConnect®

O formato das comunicações que o Agente envia e recebe está descrito em um documento XML intitulado *Devices*. O CNC (*Device*) pode ser visto como um conjunto de componentes (*Components*) – eixos, por exemplo – e esses componentes podem gerar dados (*DataItems*) ou ter sub-componentes. A estrutura do arquivo *Devices.xml* (Figura 2) procura mapear a mesma estrutura do dispositivo real, descrevendo seus componentes e os dados que cada um pode fornecer ao agente, além de conter outras informações como o nome do dispositivo monitorado e a taxa de leitura (*sampleRate*). Qualquer tentativa de enviar para o Agente um dado que não esteja descrito nesse documento resulta em uma resposta de erro à requisição *HTTP*. Existe um número limitado de categorias de componentes e tipos de dados permitidos nas versões iniciais do padrão MTConnect, mas esse número tende a aumentar conforme o padrão for sendo aceito pela indústria e mais instituições participarem da sua definição. Assim, de forma a se encaixar no padrão MTConnect o arquivo *Devices.xml* deve ser escrito conforme instruções contidas em Sobel (2009a, 2009b, 2009c).

```
- <Devices>
- <Device sampleRate="10.0" name="Mori Seiki Fanuc30i" iso841Class="1" uuid="mori-sei
  <Description manufacturer="Company Name" serialNumber="Serial Number" />
- <DataItems>
  <DataItem category="EVENT" id="id1" name="alarm" type="ALARM" />
</DataItems>
- <Components>
- <Axes name="Axes" id="id105">
- <Components>
- <Spindle name="S" id="id101">
- <DataItems>
  <DataItem category="SAMPLE" id="id3" name="Srpm" type="SPINDLE_SPEE
</DataItems>
</Spindle>
- <Linear name="X" id="id102">
- <DataItems>
  <DataItem category="SAMPLE" id="id4" name="Xabs" type="POSITION" unit
</DataItems>
</Linear>
- <Linear name="Y" id="id103">
- <DataItems>
  <DataItem category="SAMPLE" id="id6" name="Yabs" type="POSITION" unit
</DataItems>
</Linear>
- <Linear name="Z" id="id104">
- <DataItems>
```

Figura 2 – Fragmento do arquivo Devices.xml.

## 3. MATERIAIS E MÉTODOS

Para o desenvolvimento deste projeto, foi utilizada a arquitetura mostrada na Figura 3. A máquina-ferramenta monitorada foi o centro de usinagem marca Mori Seiki, modelo NVD1500 DCG. A comunicação entre o CNC Fanuc modelo 30i e o Adaptador é feita por meio da biblioteca *FOCAS2*, proprietária, a qual foi adquirida do fabricante do CNC para comunicação com PCs. O Adaptador se comunica com o Agente por meio de *HTTP Requests*, respondidos por *XML Acknowledgments*. O Cliente também envia *HTTP Requests* ao Agente, recebendo as informações requisitadas no formato XML.

O CNC foi acessado por meio de uma conexão Ethernet à rede do laboratório OPF-NUMA. Apesar de a biblioteca *FOCAS2*, utilizada para a comunicação com a máquina, também oferecer suporte a conexões do tipo HSSB, mais rápidas, a conexão Ethernet se mostrou mais prática pela possibilidade de ser integrada a uma rede, possibilitando que vários computadores tenham acesso à máquina.

Apesar de os *softwares* Adaptador e Agente possuírem a capacidade de operar em computadores diferentes através do protocolo HTTP, eles foram, por uma questão de praticidade, instalados no mesmo computador, conectado à rede mencionada anteriormente. As aplicações Clientes, no entanto, foram executadas em computadores diferentes, inclusive fora da rede em que o Adaptador e o Agente foram instalados.

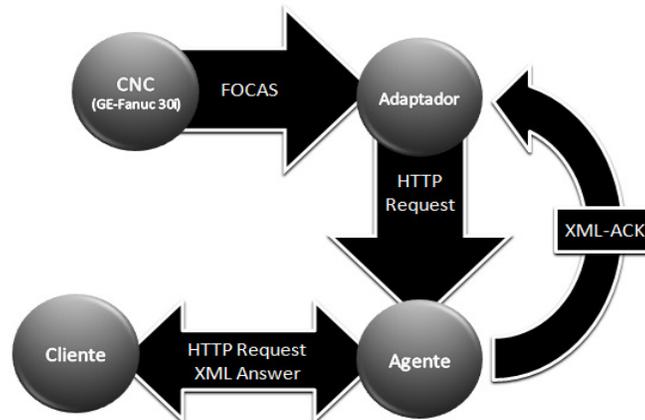


Figura 3 – Arquitetura de transferência de dados utilizada no projeto

Um esquema da arquitetura de *hardware* utilizada pode ser visto na Figura 4:

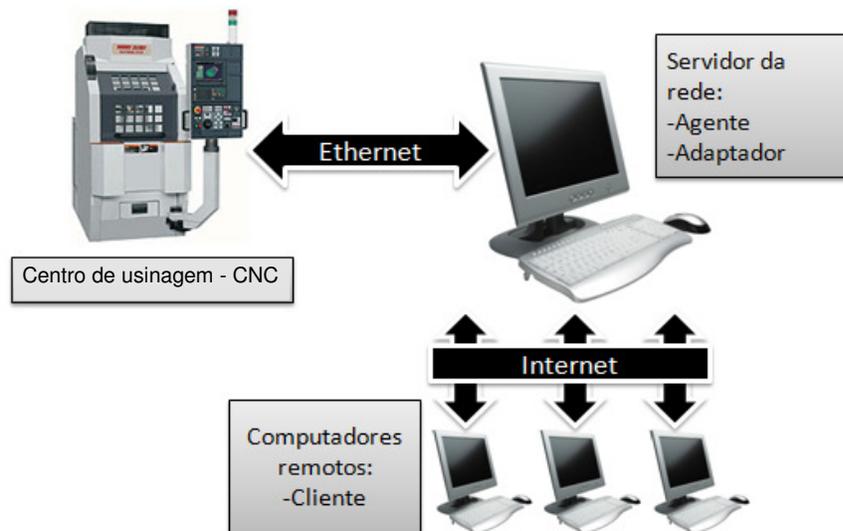


Figura 4 – Arquitetura de hardware utilizada no projeto.

### 3.1. Comunicação com a controladora CNC

Para a comunicação com o CNC, foi utilizada a biblioteca proprietária *FOCAS2* (Fanuc Open CNC API Specifications 2), da Fanuc. A leitura da documentação referente a essa biblioteca revelou um código escrito na linguagem C, otimizado para execução rápida ocupando a menor quantidade de memória possível, mas pouco prático para o uso na programação de softwares mais complexos.

Devido ao cada vez maior barateamento do poder computacional, códigos como o da biblioteca *FOCAS2* podem ser substituídos por equivalentes mais simples, sacrificando parte da eficiência computacional em favor de um código legível e que possa ser facilmente usado no desenvolvimento de vários aplicativos, sem a necessidade da leitura de pormenores de uma documentação extensa. Assim, o primeiro passo no desenvolvimento foi encapsular o código original da biblioteca *FOCAS2*, escrito em C, em funções mais simples escritas na linguagem C#.

### 3.2. Encapsulamento das funções

O encapsulamento das funções da biblioteca *FOCAS2* foi feito com base na biblioteca *fwlib32.dll*, da empresa Fanuc. As chamadas de funções dessa biblioteca são feitas originalmente utilizando algumas estruturas customizadas, além de diversos parâmetros específicos e por referências. As funções utilizadas para leitura dos dados necessários para o sistema são inicialmente importadas para a linguagem C# através da classe *Focas1* e depois encapsuladas em métodos mais simples na classe *FocasInterface*. A Tabela 1 mostra algumas das funções encapsuladas. Dessa forma, as chamadas dos novos métodos passam a ser mais intuitivas e com um número reduzido de parâmetros, criando uma camada que abstrai as chamadas em baixo nível das funções da *DLL*.

Tabela 1: Encapsulamento das funções da biblioteca *fwlib32.dll*

Funções da biblioteca <i>fwlib32.dll</i>	Métodos da classe <i>FocasInterface</i>
FWLIBAPI short WINAPI cnc_allclibhndl3(const char *ipaddr, unsigned short port, long timeout, unsigned short *FlibHndl)	public void Connect()
FWLIBAPI short WINAPI cnc_freelibhndl(unsigned short FlibHndl)	public void Disconnect()
FWLIBAPI short WINAPI cnc_rdspmeter(unsigned short FlibHndl, short type, short *data_num, ODBSPLOAD *loadmeter)	public double getSpindleLoad()
FWLIBAPI short WINAPI cnc_absolute2( unsigned short FlibHndl, short axis, short length, ODBAXIS *absolute )	public double getAbsolutePositions()
FWLIBAPI short WINAPI cnc_statinfo(unsigned short FlibHndl, ODBST *statinfo)	public double getCncExecution()

Esse processo torna a reutilização de tais métodos em outros projetos extremamente viável, uma vez que sua estrutura passa a ser familiar para qualquer desenvolvedor de linguagens orientadas a objeto. Além disso, a abstração criada facilita futuras alterações ou atualizações de código. Se outra versão da biblioteca do fabricante tiver chamadas de funções diferentes da utilizada, basta alterar tais chamadas nos métodos da classe *FocasInterface*, que as encapsulam. Essas alterações não são percebidas no restante do código, que continuará recebendo o resultado da chamada da mesma forma, independente de como a comunicação é implementada. A Figura 5 mostra o diagrama de classes do adaptador implementado. A classe *IO* implementa o método que utiliza as chamadas da classe *FocasInterface* para comunicação com o CNC e leitura dos dados (*getData*) e o método que envia os dados lidos para o agente, enquanto a classe *ThreadControl* controla a leitura e escrita dos dados na taxa configurada. As classes *AdapterException*, *AgentException* e *PowerOffException* implementam mensagens de erro customizadas para cada parte do sistema.

### 3.3. Conexão com a controladora CNC

Uma vez estabelecida a arquitetura, a biblioteca FOCAS2 se encarrega de utilizar a conexão *Ethernet* para a comunicação com a máquina CNC. No entanto, para o perfeito funcionamento da comunicação, as condições abaixo devem ser observadas:

- O CNC e o computador que executa o Adaptador devem estar conectados à mesma rede.
- O endereço de IP do CNC deve ser conhecido, juntamente com o número de uma porta válida para o acesso. Esses parâmetros podem ser configurados diretamente na controladora CNC.
- O CNC deverá estar ligado e em condições normais de funcionamento.

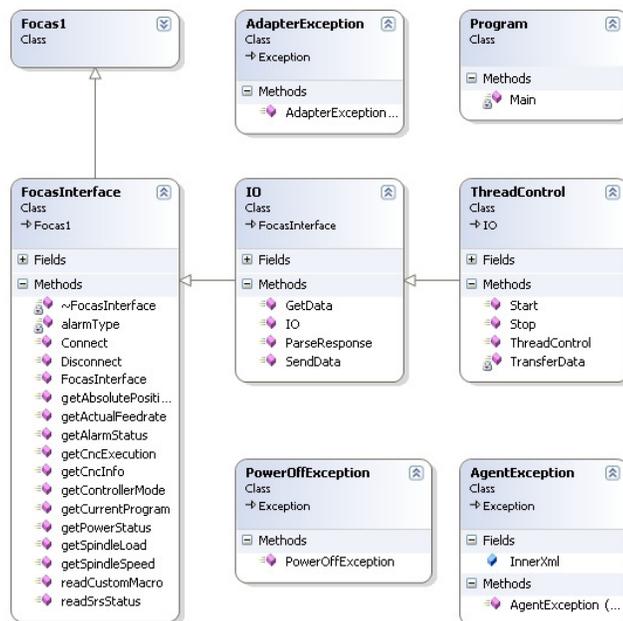


Figura 5 – Diagrama de classes do adaptador.

### 3.4. Conexão com o Agente

Para a troca de dados entre Adaptadores e o software Agente, o protocolo HTTP é utilizado. A cada novo dado coletado do CNC, um HTTP *Request* é enviado para o Agente, contendo o nome da máquina na qual o dado foi

coletado, seu valor, data e hora da coleta e quaisquer outras informações pertinentes, como a mensagem de um alarme. O Agente armazena esses dados, que podem ser utilizados posteriormente por outro aplicativo. Esse procedimento é repetido para cada amostra de cada um dos dados coletados (Figura 6).

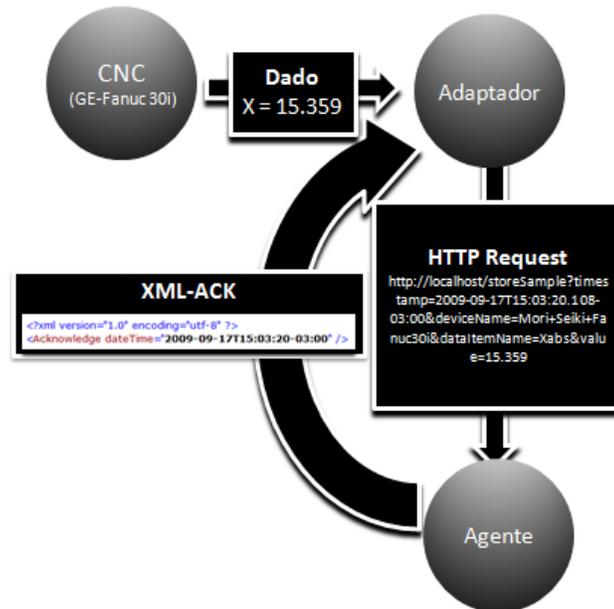


Figura 6 – Trajetória de uma amostra coletada da controladora ao Agente.

### 3.5. O Adaptador

O Adaptador desenvolvido coleta informações do CNC da máquina-ferramenta, tais como: coordenadas absolutas dos eixos da máquina (*Xabs*, *Yabs* e *Zabs*), em milímetros, a rotação do eixo árvore ou *spindle* (*Srpm*), em rotações por minuto, avanço atual [mm/min], carregamento no eixo árvore, dados do CNC e alarmes, na taxa de amostragem definida pelo usuário.

Para a primeira versão do Adaptador foi criada uma interface de usuário simples e amigável, com a intenção de facilitar o uso (Figura 7):

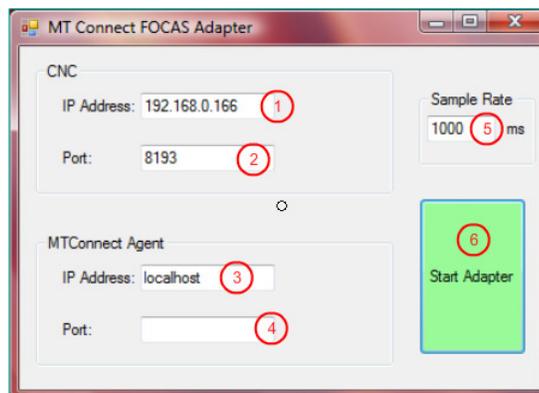


Figura 7 – Interface de usuário do Adaptador, primeira versão.

A configuração do Adaptador pode ser feita de acordo com as instruções:

1. Endereço de IP do CNC – Deve-se fornecer um endereço de IP válido para o CNC Fanuc 30i.
2. Porta da CNC – Deve-se fornecer uma porta válida para comunicação com o CNC.
3. Endereço de IP do Agente – Deve-se fornecer um endereço de IP válido para o computador onde o software Agente está sendo executado. Caso o Agente esteja sendo executado no mesmo computador que o Adaptador, *localhost* pode ser usado neste campo.
4. Porta do Agente – Deve-se fornecer uma porta válida para comunicação com o Agente. Caso o Agente esteja sendo executado no mesmo computador que o Adaptador, este campo pode ser deixado em branco.
5. Taxa de amostragem – Deve-se fornecer a taxa de amostragem desejada, em milissegundos. Valores acima de 60 são recomendados para melhor desempenho.
6. Botão de início/parada – Início/interrupção do adaptador.

A interface da primeira versão do Adaptador, embora de fácil entendimento e uso, mostrou não ser prática ao longo do tempo. A necessidade de mudar manualmente na interface as configurações a cada nova execução do Adaptador tornou-se um problema durante os testes do programa, que exigem que o mesmo seja aberto e fechado várias vezes para mudanças no código. Por isso, no desenvolvimento da segunda versão do Adaptador a interface foi alterada, mudando seu foco de facilidade de uso para praticidade.

A segunda versão do Adaptador mostra ao usuário apenas uma janela de console, onde são mostradas mensagens referentes à inicialização do Adaptador e a erros e advertências que ocorram durante sua execução. A coleta de dados é iniciada automaticamente quando o software é inicializado (Figura 8).

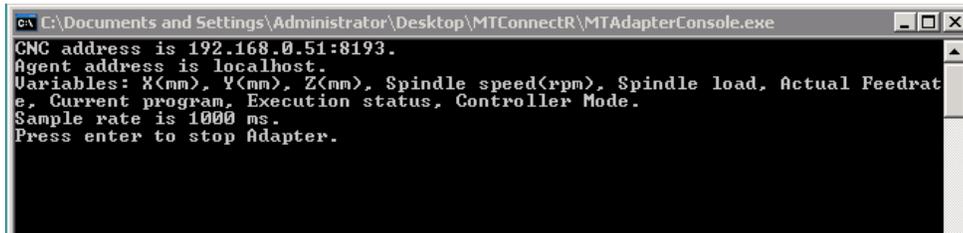


Figura 8 – Interface de usuário do Adaptador, segunda versão.

Para a configuração, bem mais flexível que na primeira versão do adaptador, é utilizado um arquivo no formato XML, com o mesmo nome do executável do Adaptador e de extensão “.config” (Figura 9). As principais configurações encontradas no arquivo estão descritas abaixo:

- *CncIP*, *CncPort*, *AgentIP*, *AgentPort*, *SampleRate* - Os valores e as regras para o preenchimento são os mesmos descritos na interface de usuário na primeira versão do Adaptador.
- *deviceName* - Permite que o usuário escolha o nome da máquina utilizada. Com a correta configuração deste parâmetro no Adaptador e no Agente, vários Adaptadores podem ser utilizados simultaneamente para o envio de dados sobre diferentes máquinas para o mesmo Agente. Caso o nome contenha espaços, eles devem ser substituídos por “+”.
- “get...” – Permite a escolha das variáveis que serão coletadas pelo Adaptador. Cada parâmetro deve ser definido como *true* ou *false*. Por exemplo, definir o parâmetro *getXabs* como *true* faz com que o programa colete informações sobre a posição absoluta do eixo X.

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <appSettings>
    <!-- Address and Sample Rate configurations-->
    <add key="CncIP" value="192.168.0.51" />
    <add key="CncPort" value="8193" />
    <add key="AgentIP" value="localhost" />
    <add key="AgentPort" value="" />
    <add key="SampleRate" value="1000" />
    <!-- Device Name-->
    <add key="deviceName" value="Mori+Seiki+Fanuc+30i" />
    <!-- Variable configurations-->
    <!-- Absolute X, Y, Z coordinates-->
    <add key="getXabs" value="true" />
    <add key="getYabs" value="true" />
    <add key="getZabs" value="true" />
    <!-- Spindle Speed and Load-->
    <add key="getSrpm" value="true" />
    <add key="getSload" value="true" />
    <!-- Actual Feedrate-->
    <add key="getActf" value="true" />
    <!-- Program being currently executed-->
    <add key="getCurrProg" value="true" />
    <!-- Execution status (start, stop, pause, ...)-->
    <add key="getExecution" value="true"/>
    <add key="getContMode" value="true"/>
    <!-- END of variable configurations-->
    <!-- END OF CONFIGURATIONS-->
  
```

Figura 9 – Fragmento do arquivo de configuração da segunda versão do Adaptador

### 3.6. O Agente

Para a comunicação com o adaptador, foi utilizado a agente padrão disponibilizado pelo MTConnect Institute. A inicialização do mesmo ocorre através da interface apresentada na Figura 10.



Figura 10 – Tela de Inicialização do Agente

#### 4. RESULTADOS E DISCUSSÃO

O padrão MTConnect utiliza o formato XML como forma de organização dos dados. Entretanto, o XML fornece apenas a estrutura dos mesmos, ou seja, não contém nenhuma informação sobre como esses dados devem ser apresentados. Essa separação entre conteúdo e apresentação permite que os mesmos dados possam ser utilizados em diferentes aplicativos. Para definir o modo de apresentação dados descritos em XML pode ser usado a *Extensible Style Language* (XSL) e a *Cascading Style Sheets* (CSS).

Assim, os dados em formato XML são lidos do agente MTConnect através de uma requisição HTTP, e uma transformação XSL gera a página HTML para apresentação no *browser*. A transformação XSL permite formatar dinamicamente a página acrescentando ou removendo campos, formatando cores e fontes, de acordo com a estrutura dos dados XML.

A página web desenvolvida é mostrada na Figura 11. Nela são visualizados os dados referentes ao status do CNC e dados do mesmo. O avanço atual, a rotação do eixo e carregamento do mesmo são informações apresentadas. As posições atuais dos eixos X, Y e Z estão no box “*Actual Position*”. Informações adicionais do controle, bem como o programa em execução são mostrados no box *Controller*.

Machine Information	
Power Status	ON
Device Name	Mori Seiki Fanuc 30i
Actual Feedrate	0
Spindle Speed	0
Spindle Load	0
Actual Position	
X	1.0024
Y	157.2135
Z	144.4166
Controller	
Controller Mode	MANUAL_DATA_INPUT
Execution	STOPPED
Current Program	O0

Figura 12 – Tela do aplicativo cliente

#### 5. CONCLUSÕES

- A utilização do padrão MTConnect permitiu o desenvolvimento de uma interface de comunicação com a máquina-ferramenta em questão, através do desenvolvimento dos diferentes elementos que compõem a estrutura do padrão.
- O adaptador desenvolvido permitiu a coleta dos dados da máquina-ferramenta remotamente via Ethernet;
- A conexão adaptador-agente foi estabelecida com sucesso, permitindo a geração do arquivo de saída XML com as informações pertinentes da máquina-ferramenta;
- Com o aplicativo cliente proposto foi possível coletar as informações de saída do arquivo XML e gerar a página HTML para apresentação no browser.
- O sucesso no desenvolvimento das etapas para implementação do padrão e na coleta dos dados permitirá que novas aplicações clientes sejam desenvolvidas, incluindo agora diferentes máquinas-ferramentas com CNC de fabricantes distintos.

#### 6. AGRADECIMENTOS

Os autores gostariam de agradecer os seguintes institutos e empresas pelo apoio a esta pesquisa: CNPq, CAPES, Mori Seiki Co. e AMT.

## 7. REFERÊNCIAS

- Ferraz Júnior, F., 2007 “Arquiteturas para monitoramento e supervisão integrados de processos de usinagem em máquinas com controle numérico aberto”. Escola de Eng. de São Carlos, Univ. de São Paulo, Tese de Doutorado.
- Difrank, G., 2008, “Power of automation - An overview, technology, and implementation”, IEEE Industry Applications Magazine, Mar-Apr, pp. 49-57;
- IETF, 2010, “HTTP/1.1 specifications - HTTP/1.1, part 1: URIs, Connections, and Message Parsing” <http://www.ietf.org/id/draft-ietf-httpbis-p1-messaging-09.txt>
- Oliveira, J. F. G.; Ferraz Júnior, F.; Coelho, R. T.; Silva, E. J., 2008, An Innovative architecture for machining process and production monitoring. Journal of Engineering Manufacture –Proceedings of the Institution of mechanical engineers Part B, v.222, pp. 1605-1612.
- Rockwell, 2004, “Making Sense of e-Manufacturing: A Roadmap for UK Industry” [http://www.rockwellautomation.co.uk/applications/gs/emea/GSGB.nsf/files/emanwp.pdf/\\$file/emanwp.pdf](http://www.rockwellautomation.co.uk/applications/gs/emea/GSGB.nsf/files/emanwp.pdf/$file/emanwp.pdf)
- Sauter, T., 2005, “Integration aspects in automation – a technology survey”, Proc. of the 10th IEEE Int. Conf. on Emerging Technologies and Factory Automation (ETFA'2005), Vol. 2, pp. 255–263.
- Sobel, W., 2009a. “MTConnect Standard Part 1 – Overview and Protocol”. Versão 1.0.1. Disponível em: <[http://mtconnect.org/images/documents/standard/MTC\\_Part\\_1\\_Overview\\_v1.0.1.pdf](http://mtconnect.org/images/documents/standard/MTC_Part_1_Overview_v1.0.1.pdf)>. Acesso em: 17 Set. 2009.
- Sobel, W., 2009b. “MTConnect Standard Part 2 – Components and Data Items”. Versão 1.0.1. Disponível em: <[http://mtconnect.org/images/documents/standard/MTC\\_Part\\_2\\_Components\\_v1.0.1.pdf](http://mtconnect.org/images/documents/standard/MTC_Part_2_Components_v1.0.1.pdf)>, Acesso em: 17 Set. 2009.
- Sobel, W., 2009c. “MTConnect Standard Part 3 – Streams, Events and Samples”. Versão 1.0.1. Disponível em: <[http://mtconnect.org/images/documents/standard/MTC\\_Part\\_3\\_Streams\\_v1.0.1.pdf](http://mtconnect.org/images/documents/standard/MTC_Part_3_Streams_v1.0.1.pdf)>. Acesso em: 17 Set. 2009.
- Warndorf, P., Fox, A., Sobel, W., 2007. “MTConnect Technology White Paper”. Disponível em: <<http://mtconnect.org/media/MTConnectWhitePaper.doc>>. Acesso em: 17 Set. 2009

## 8. DIREITOS AUTORAIS

Os autores são os únicos responsáveis pelo conteúdo do material impresso incluído no seu trabalho.

# DEVELOPMENT OF A MONITORING SYSTEM FOR MACHINING TOOLS USING THE MTCONNECT®

Eraldo Jannone da Silva, [eraldojs@sc.usp.br](mailto:eraldojs@sc.usp.br)<sup>1</sup>  
Leandro Coelho Rondon, [rondonusp@gmail.com](mailto:rondonusp@gmail.com)<sup>2</sup>  
Mariana Vilela Martins, [mariana.mk07@gmail.com](mailto:mariana.mk07@gmail.com)<sup>2</sup>  
Thiago Valle França, [thiagovf@sc.usp.br](mailto:thiagovf@sc.usp.br)<sup>1</sup>  
José Eduardo Cervelin, [cervelin@usp.br](mailto:cervelin@usp.br)<sup>1</sup>

<sup>1</sup>University of São Paulo - USP – School of Engineering- São Carlos - EESC; Production Engineering Department – SEP. Av. Trabalhador São-carlense, 400, CEP: 13566-590, São Carlos – SP, Brazil.

<sup>2</sup>University of São Paulo - USP - Institute of Mathematics and Computer Science. Av. Trabalhador São-carlense, 400, CEP: 13560-570, São Carlos – SP, Brazil.

**Abstract.** *With the development of NC machine tools, the real-time data collection from the process is a competitive advantage to those adopting that strategy. The data acquisition in production lines with high number of machines with different NC manufactures is still a great challenge, considering that the customization of monitoring and supervisory system is necessary, due to the distinct and proprietary characteristic of the communication protocols. The MTConnect protocol was created to become a standard for machine tools and NC manufactures, as the Bluetooth and the USB are in their respective segments. By the adoption of a unique data exchange format, the implementation of systems for analysis and control can be easily achieved. The current machine tools don't support the MTConnect standard. The development of an adapter is necessary to “translate” the information to the standard. The adapter can be installed in the server computer or in the NC itself (for open NC controller). Furthermore, the standard uses HTTP protocol to data exchange, allowing the monitoring of simultaneous machines. The aim of this study was to develop a monitoring system using a MTConnect protocol for on-line data collection of the main parameters of a machine center, such as axes actual position, spindle rpm, spindle load, among others. The architecture elements were developed, including an adapter for Fanuc NC. As a result, it was possible to collect through HTTP the main process parameters.*

**keywords:** machining, monitoring, NC