UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Robson Neves Gonçalves

Desenvolvimento de Servidores OPC DA, OPC UA e Wrappers para aplicação em Automação.

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica como parte dos requisitos para obtenção do Título de ***Mestre em Ciências em Engenharia Elétrica***

Área de Concentração: Automação e Sistemas Elétricos Industriais.

Orientador(a): Profª. Lúcia R. H. Franco, Dra.

Fevereiro de 2012

Itajubá - MG

Ficha catalográfica elaborada pela Biblioteca Mauá –
Bibliotecária Cristiane N. C. Carpinteiro- CRB_6/1702

G635d

Gonçalves, Robson Neves
Desenvolvimento de servidores OPC DA, OPC UA e wrappers para Aplicação em automação . / por Robson Neves Gonçalves. -- Itajubá (MG) : [s.n.], 2012.

82 p. : il.

Orientadora: Profa. Dra. Lúcia Regina Horta Rodrigues Franco.
Dissertação (Mestrado) – Universidade Federal de Itajubá.

1. OPC UA. 2. OPC DA. 3. Wrappers OPC. I. Franco, Lúcia Regina Horta Rodrigues, orient. II. Universidade Federal de Itajubá. III. Título.

UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Robson Neves Gonçalves

Desenvolvimento de Servidores OPC DA, OPC UA e Wrappers para aplicação em Automação.

Dissertação aprovada por banca examinadora em 29/02/2012 , conferindo ao autor o título de ***Mestre em Ciências em Engenharia Elétrica***

Banca Examinadora

Profª. Lúcia R. H. Franco, Dra. (Orientadora)

Prof. Luiz Gabriel Lenarth, Dr.

Profª. Itana Stiubiener, Dra.

Fevereiro de 2012

Itajubá - MG

Robson Neves Gonçalves

Desenvolvimento de Servidores OPC DA, OPC UA e Wrappers para aplicação em Automação.

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica como parte dos requisitos para obtenção do Título de ***Mestre em Ciências em Engenharia Elétrica***

Aprovada em 29 de Fevereiro de 2012.

COMISSÃO EXAMINADORA

_____

Profª. Lúcia R. H. Franco, Dra. (Orientadora)

_____

Prof. Luiz Gabriel Lenarth, Dr.

_____

Profª. Itana Stiubiener, Dra.

RESUMO

Este trabalho apresenta as informações relevantes para o desenvolvimento de servidores OPC DA, UA e Wrappers. Aborda o conceito teórico sobre a tecnologia e as ferramentas utilizadas para o desenvolvimento, proporcionando a síntese para a elaboração desse tipo de sistema. O estudo é contextualizado em uma situação real de aquisição de dados usando o sistema supervisório Elipse E3, para supervisionar os resultados provenientes do servidor OPC DA/UA e o Elipse Plant Manager - EPM para aquisitar os resultados provenientes do servidor OPC DA (Elipse E3), e ser entendido por um OPC UA Cliente através do OPC Wrapper. Para o Ambiente de Teste foi configurada uma rede MODBUS TCP, com os módulos de Automação da empresa Advantech – ADAM. Para garantir a conformidade dos servidores criados são realizados os testes de conformidade, utilizando o Compliance Test Tool – CTT da OPC Foundation, demonstrando a capacidade de interoperabilidade fornecida pelas soluções desenvolvidas. Após os testes são apresentadas os principais pontos de falhas ocorridas e as respectivas soluções.

Palavras-chave: OPC UA, OPC DA, *Wrappers OPC*, Elipse E3, Elipse Plant Manager, Sistemas Supervisórios, SCADA, CTT.

ABSTRACT

This work presents relevant information to the development of OPC servers (DA and UA) and OPC Wrappers. It provides the concepts needed to develop this type of system. The study is realized in a real time data acquisition by the E3 software, supervising the results from the OPC DA server / UA and Elipse Plant Manager – EPM to give the results from the OPC DA (E3), and be understood by an OPC Client through the Wrapper OPC. For the test environment was set up a Modbus TCP network with modules from Advantech - ADAM Automation Company. To ensure the compliance of servers created are performed compliance testing using the Compliance Test Tool - CTT of the OPC Foundation, demonstrating the ability of interoperability provided by the solutions developed- Also are presented the results of Compliance tests, demonstrating the ability to interoperability provided by the implemented solutions. After the tests are presented the main points of failure that occurred and their solutions.

Key-words: OPC UA, OPC DA, Wrappers OPC, Elipse E3, Elipse Plant Manager, Supervisory Systems – SCADA.

LISTA DE SIGLAS E ABREVIATURAS

OPC	*OLE for Process Control*
SCADA	*Supervisory Control and Data Acquisition*
COM	*Component Object Model*
DCOM	*Distributed Component Object Model*
CLP	Controladores Lógico Programáveis
DCS	*Distributed Control Systems*
MES	*Manufacturing Execution Systems*
RTS	*Real Time Systems*
PC	*Personal Computer*
OSI	*Open Systems Interconnection*
TCP	*Transmission Control Protocol*
IP	*Internet Protocol*
IHM	*Interface Homem Máquina*
DOS	*Disk Operating System*
UA	*Unified Architecture*
ERP	*Enterprise Resource Planning*
PC	*Personal Computer*
OSF	*Open Software Foundation*
DCE	*Distributed Computing Environment*
RPC	*Remote Procedure Call*
EJB	*Enterprise Java Bean*
HMI	*Human Machine Interface*
DA	*Data Access*
A&E	*Alarms & Events*
HDA	*Historical Data Access*
EPM	*Elipse Plant Manager*
AC	*Alarm and Condition*
FDI	*Field Device Integration*
EDDL	*Eletronic Device Description Language*
FDT	*Field Device Tool*

HA	*Historical Access*
PROG	*Programs*
CTT	*Compliance Test Tool*
IEDs	*Intelligent Electronic Devices*
SDK	*Software Developing kit*
IETF	*Internet Engeneering Task Force*
ADAM	*Advantech Data Acquisition Modules*

LISTA DE FIGURAS

SUMÁRIO

1. INTRODUÇÃO

1.1 Considerações Iniciais

As regras de comunicação entre componentes distribuídos possuem papel fundamental na evolução tecnológica, otimizando o processo produtivo. Essa otimização é realizada através da consistência do fluxo de dados, qualidade e disponibilidade destes dados em nível de campo.

A utilização de interfaces padronizadas por diversos fabricantes é um pré-requisito para flexibilidade e integração de softwares e componentes.

Segundo (MAHNKE;LEITNER;DAMM,2009) o OPC (*OLE for Process Control*) tem sido aceito como uma interface padrão entre clientes e servidores, para diferentes campos de aplicação, se tornando o padrão mais popular entre usuários e desenvolvedores. A maioria dos fabricantes tanto de IHMs (Interface Homem Máquina), SCADA (*Supervisory Control and Data Acquisition*), e DCS (*Distributed Control System*) quanto serviços de controle, MES (*Manufacturing Execution Systems*) e ERP (*Enterprise Resource Planning*) oferecem interfaces OPC Cliente e/ou Servidor.

O Padrão OPC surgiu da necessidade de se ter uma interface de comunicação única entre um incontável número de redes, protocolos proprietários e interfaces usadas. Esse padrão reuni uma série de vantagens aos sistemas industriais, que os diferenciam dos demais, como: o isolamento de tráfego, a facilidade de integração entre sistemas e a interoperabilidade (NASCIMENTO FILHO,2005).

Com isto, a aquisição de informações não fica dependente de um driver específico, desenvolvido por um fornecedor de uma determinada solução. O

uso de um padrão único para a execução da comunicação entre o campo e os sistemas de supervisão e gerenciamento faz com que os custos de desenvolvimento e manutenção de sistemas caiam substancialmente. O conhecimento necessário para se trabalhar com as tecnologias envolvidas no desenvolvimento de soluções em um padrão único não é demasiadamente grande, e sua expansão não depende de novas curvas de aquisição de conhecimento (RIEDL,2001).

Atualmente, uma nova versão do Padrão OPC, o *OPC UA* (*Unified Architecture*), está sendo comercializada no mercado. Este novo padrão foi desenvolvido com base na experiência adquirida desde a implantação da Tecnologia OPC, mais de treze anos desde o seu início, e mudanças tecnológicas ocorridas durante esse período.

Segundo, (BURKE;IWANITZ;LANGE,2010), algumas razões motivaram o desenvolvimento dessa nova geração, citando: descontinuação do COM/DCOM (*Component Object Model / Distributed COM*), limitações DCOM, utilização do OPC em plataformas Não-Windows, necessidade de Alta Performance em comunicação via Web Services e a integração de todas as ferramentas em um modelo unificado.

OPC UA fornece estratégias de migração para o OPC UA. Ferramentas como *Wrappers* and *Proxies* fornecidos pela *OPC Foundation* estão disponíveis para traduzir diferentes interfaces do OPC Clássico para o OPC UA e vice-versa.

1.2 Motivação

A principal motivação de se trabalhar com o OPC é a capacidade e possibilidade de se trabalhar com informações entre diversas tecnologias e plataformas.

Com a implantação do *OPC UA*, novas características são adicionadas ao padrão OPC, porém o *OPC UA* pode coexistir com o padrão clássico. Isso permite o acesso às novas propriedades sem alterar ou modificar a tecnologia já implantada, reduzindo o período de comissionamento e start-up, possibilitando uma migração suave entre as duas tecnologias.

Atualmente, milhares de produtos utilizam o OPC Clássico, e podem também utilizar as vantagens do *OPC UA*. Esta migração pode ser realizada através do desenvolvimento de *Wrappers* (Acesso Servidor COM de um Cliente UA) e *Proxies* (Acesso Servidor UA de um Cliente COM). Neste ponto, o desenvolvimento dessas ferramentas torna-se um diferencial no mercado de Automação e Redes Industriais.

Segundo (NASCIMENTO FILHO,2005), este padrão por ser Internacionalmente conhecido e desenvolvido por grandes empresas desenvolvedoras de sistemas de automação industrial, tornou-se realmente um padrão de baixa mutabilidade e fácil conformidade.

1.3 Objetivos

1.3.1 *Objetivo Geral*

Desenvolver dois servidores OPC DA e OPC UA baseado nas especificações da *OPC Foundation* e um *Wrapper* para realização de Acesso a um Servidor COM através de um Cliente UA.

1.3.2 Objetivos Específicos

- Realizar um estudo sobre a tecnologia do OPC Clássico e OPC UA.
- Apresentar informações relevantes para o desenvolvimento de servidores OPC DA e UA.
- Apresentar um sistema de comunicação confiável para aquisição de informações disponibilizando-as de forma padronizada.
- Desenvolver um *Wrapper* que seja capaz de intermediar a comunicação entre as duas tecnologias.

1.4 Organização do documento

O capítulo 1 é responsável por toda parte introdutória do trabalho, incluindo a motivação, o objetivo geral e os objetivos específicos.

O capítulo 2 abrange toda a revisão de literatura relacionada à OPC Foundation, o OPC Clássico (OPC DA, OPC A&E, OPC HDA), incluído também a ferramenta para análise de conformidade dos Servidores, o Compliance test Tool - CTT.

O capítulo 3 compreende toda a revisão de literatura relacionada ao

OPC UA, incluído uma visão geral sobre a tecnologia, as especificações e também a ferramenta para análise de conformidade do Servidores UA, o “Compliance test Tool OPC UA “– CTT UA.

O capítulo 4 é responsável pela revisão de literatura relacionada à Migração do OPC Clássico para o OPC UA e vice-versa, focando principalmente nos “Wrappers” e “Proxies”.

O capítulo 5 abrange todo o conteúdo técnico relacionado aos sistemas supervisórios, indo do conceito teórico até os softwares utilizados no trabalho (E3 e EPM), dando uma visão geral sobre o desenvolvimento de projetos nesta área.

O capítulo 6 compreende o conceito de redes industriais, com um foco maior no MODBUS TCP, sendo a rede utilizada no trabalho.

O capítulo 7 é responsável por toda a parte prática e desenvolvimento do trabalho, focando no desenvolvimento dos servidores através dos toolkits, o ambiente de desenvolvimento, os testes dos servidores, testes de conformidade e com clientes.

O capítulo 8 abrange o desenvolvimento do Wrapper OPC, principalmente com relação ao ambiente de desenvolvimento, de testes e os resultados, além de compreender a utilização do Elipse Plant Manager – EPM.

O capítulo 9 apresenta a conclusão do trabalho, com todos importantes relacionados ao trabalho.

2. OPC

2.1. OPC Foundation

O uso do PC e softwares em sistemas de automação industrial rapidamente cresceu desde os anos 90, especialmente PCs com sistemas operacionais Windows, usados para visualização e controle de processos. Nos últimos anos um dos maiores esforços para o desenvolvimento de uma padronização de softwares de automação, foi a padronização do acesso aos dados em equipamentos onde diversos sistemas, protocolos, e interfaces eram usados.

Problemas para aplicações de software já existiam para o acesso a impressoras em ambiente DOS. Todas as aplicações precisavam inserir seus próprios *drivers* de impressora para todas as impressoras suportadas, um trabalho árduo e complicado na época. O Windows resolveu o problema dos drivers, incorporando o suporte a impressoras no sistema operacional. Uma interface com os *drivers* de impressora servia todas as aplicações que precisavam de acesso à impressora.

Os vendedores de softwares de IHMs e SCADA tinham problemas similares, e uma força tarefa foi iniciada com o objetivo de definir um padrão de drivers para o Plug&Play de equipamentos, permitindo um acesso padronizado aos dados de Automação sobre plataforma Windows.

O resultado foi à criação de uma especificação para o *OPC Data Access*, em Agosto de 1996. Neste mesmo período foi criada a *OPC Foundation*, uma organização sem fins lucrativos que tem como objetivo manter e desenvolver este padrão (MAHNKE;LEITNER;DAMM,2009).

2.2. OPC Clássico

Nos últimos anos, a *OPC Foundation* tem definido interfaces de softwares para padronizar o fluxo de informação no nível de campo até o nível de gerenciamento. De acordo com a *OPC COMMON DEFINITIONS AND INTERFACES* (1998), com as diferentes necessidades dentro das aplicações industriais, três principais especificações OPC foram desenvolvidas inicialmente: *Data Access* (DA), *Alarm & Events* (AE) e *Historical Data Access* (HDA).

O OPC usa uma arquitetura cliente-servidor para a troca de informações. Um Servidor OPC encapsula as informações de processo e disponibiliza na sua interface. Um OPC Cliente conecta no Servidor OPC e pode acessar e consumir os dados oferecidos. As aplicações consomem e fornecem dados que podem ser tanto do cliente quanto do servidor (Figura 1).

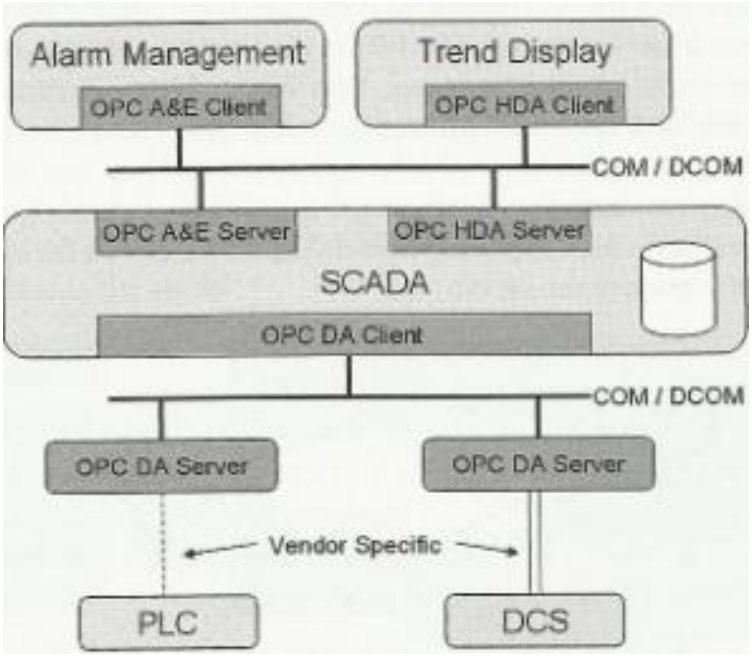


Figura 1 - Caso Típico comunicação Clientes e Servidores OPC

Fonte: MAHNKE;LEITNER;DAMM,2009

Interfaces de OPC Clássico são baseadas na tecnologia COM e DCOM da Microsoft. A vantagem dessa parceria foi à redução do trabalho inicial de especificação para definição de diferentes APIs para diferentes necessidades de cada especialidade, sem a necessidade de definir um protocolo de rede ou um mecanismo para comunicação entre os processos.

O COM e DCOM fornecem um mecanismo transparente para um cliente chamar métodos sobre um objeto COM, em um servidor rodando em um mesmo processo, em outro processo, ou em outro nó na rede. (MAHNKE;LEITNER;DAMM,2009)

Usando esta tecnologia disponível em todos os PCs com sistema operacional Windows reduziu-se o tempo de desenvolvimento das especificações e produtos, além de reduzir o tempo de chegada do OPC ao mercado. Esta vantagem foi importante para o sucesso do OPC.

As duas principais desvantagens são a dependência do OPC em uma plataforma Windows e os problemas da tecnologia DCOM na utilização de comunicação remota com o OPC.

A tecnologia DCOM é de difícil configuração, tem períodos longos e não configuráveis de timeouts, e não podem ser usados para comunicação internet (BURKE;IWANITZ;LANGE,2010).

2.3. OPC Data Access

A interface do *OPC Data Access* permite leitura, escrita e monitoramento das variáveis dispostas no processo. O seu principal uso é mover em tempo real dados de CLPs, DCSs e outros equipamentos de controle para IHMs e

outros supervisórios (*OPC DATA ACCESS CUSTOM INTERFACE STANDARD, 2002*).

O *OPC DA* é a mais importante interface OPC, implantada em 99% dos produtos que utilizam a tecnologia OPC atualmente (MAHNKE;LEITNER;DAMM,2009).

O Cliente OPC DA seleciona as variáveis (*OPC Items*) que quer ler, escrever ou monitorar no Servidor. O Cliente OPC estabelece a comunicação com o Servidor pela criação do objeto *OPCServer*. O Objeto *OPCServer* oferece métodos para navegar através de espaços de endereço e encontrar itens e suas propriedades, como: tipo de dados e direitos de acesso.

Para acessar os dados, os *OPC Items* possuem a mesma configuração, inclusive tempo de atualização no objeto *OPC Group*. A Figura 2 mostra os diferentes objetos que o OPC Cliente cria no servidor.

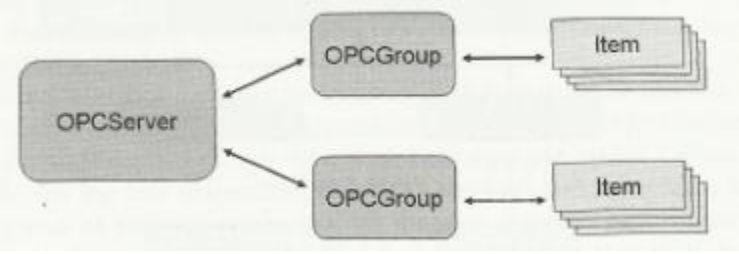


Figura 2 - Objetos Criados por um OPC Cliente para Acesso aos Dados

Fonte: MAHNKE;LEITNER;DAMM,2009

Quando adicionados a um grupo, os *items* podem ser lidos ou escritos pelo cliente. O caminho preferido para a leitura cíclica de dados pelo cliente é monitorando as mudanças de estado no servidor. O cliente define uma taxa de atualização do grupo, contendo os *items* de interesse. A taxa de atualização é usada no servidor para o check cíclico de mudança de valores/estado. Depois de cada ciclo, o servidor apenas atualiza os valores do cliente.

O *timestamp* e a qualidade na entrega dos dados são fornecidos pelo OPC Clássico. A qualidade na entrega avalia se o dado é preciso, se não está disponível (mal) ou não conhecido (Incerto) (*OPC DATA ACCESS CUSTOM INTERFACE STANDARD, 2002*).

2.4. OPC Alarm & Events

O *OPC A&E* habilita a recepção de notificações de eventos e alarmes. Eventos são simples notificações informando o cliente sobre alguma ocorrência. Os Alarmes são notificações que informam os clientes sobre as mudanças das condições de processo (*OPC COMMON DEFINITIONS AND INTERFACES,*1998).

Muitos alarmes incluem um ACK e este ACK também pode ser incluído na interface *OPC A&E*. Para receber notificações, o Cliente *OPC A&E* conecta-se ao servidor, subsescreve a notificação, e então recebe todas as notificações acionadas no servidor.

O Cliente OPC conecta-se ao Servidor A&E pela criação do objeto *OPCEventServer*, e posteriormente pelo *OPC EventSubscription*, usado para receber as mensagens de evento.

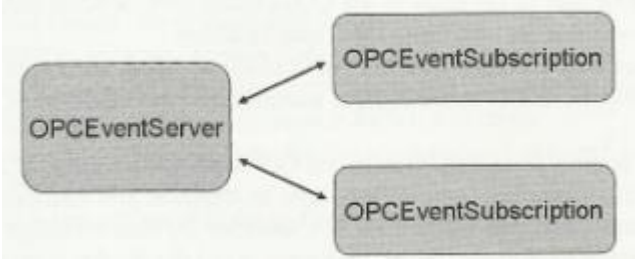


Figura 3 - Objetos criados pelo Cliente OPC para receber eventos

Fonte: MAHNKE;LEITNER;DAMM,2009

2.5. OPC Historical Data Access

Como o *OPC DA* fornece dados em tempo real, continuamente ocorrem atualizações de dados, segundo (MAHNKE;LEITNER;DAMM,2009), o *OPC Historical Data Access* dá acesso aos dados armazenados. De um simples "login" a complexos sistemas SCADA, arquivos históricos podem ser recuperados de maneira uniforme.

O Cliente OPC conecta ao Servidor HDA pela criação do objeto *OPCHDAServer*. Este objeto oferece todas as interfaces e métodos para leitura e atualização dos dados históricos. Um segundo objeto, *OPCHDABrowser*, é definido para navegação nos endereços do Servidor HDA.

A principal funcionalidade do HDA é a leitura de dados históricos de três diferentes maneiras, a primeira é realização a leitura das linhas de dados do arquivo, onde o cliente define um ou mais variáveis e o tempo que ele quer ler. O servidor retorna todos os valores arquivados conforme especificado pelo cliente.

A segunda maneira é a leitura dos valores de uma ou mais variáveis conforme *timestamps* especificados.

A terceira forma computa valores de dados agregados em uma base de dados para um especificado domínio para uma ou mais variáveis. Os valores arquivados sempre possuem associado à qualidade e *timestamp*. Além disso, nos métodos de leitura, o *OPC HDA* também define métodos para inserção, troca, e exclusão de dados em uma base de dados (MAHNKE;LEITNER;DAMM,2009).

2.6. Compliance Test Tool (OPC Clássico)

Um padrão aberto tal como o OPC é criado com o objetivo de permitir a interoperabilidade entre produtos de diferentes fabricantes. Para assegurar que o produto cumpriu com as especificações deve-se realizar uma série de testes que certificam o produto.

A não conformidade de um produto tipicamente decorre de uma interpretação incorreta da especificação. O Teste de Conformidade é projetado para auxiliar o fabricante a encontrar e corrigir não conformidades antes de encontrá-los no campo.

2.7. Auto Certificação

A auto certificação é separada em dois grupos, um para a certificação de servidores e outra para clientes. Para servidores, a *OPC Foundation* criou e tem mantido um conjunto de testes de auto certificações, o chamado Compliance Test Tool, e o disponibiliza para membros.

Os Auto Testes são projetados para verificar se um servidor segue as varias especificações que compreendem o Padrão OPC. Para clientes, a *OPC Foundation* fornece o *OPC Analyser Client Test Tool*, ferramenta que avalia o nível de certificação dos clientes.

2.8. Servidor Compliance Test Tool

A *OPC Foundation* fornece o *Compliance Test Tool* para servidores. Fabricantes de servidores realizam o download e executam estes testes para verificarem a conformidade de seus produtos. Os Testes são realizados na

forma de uma aplicação Cliente OPC que foi projetada para executar uma série de verificações pela interoperabilidade com o servidor.

Os testes são desenvolvidos para verificar o comportamento do servidor, usando parâmetros válidos quando fazem chamadas, e também parâmetros inválidos. É importante não apenas verificar se o servidor é conforme em condições normais, mas também quando o cliente não está se comportando bem.

O *Compliance Test Tool* suporta as seguintes especificações OPC:

- *OPC Alarm and Events 1.1*
- *OPC Data Access 2.05a*
- *OPC Data Access 3.0*
- *OPC Historical Data Access 1.2*
- *OPC XML Data Access 1.01*

Para cada especificação disponível o *Compliance Test Tool* suporta as seguintes possibilidades de testes OPC (OPC TEST LAB SPECIFICATION: PART 1 – CONCEPTS, 2008):

- Testes Lógicos – Esses testes checam se o *Proxy/Stub-DLLs* estão instalados corretamente.
- Interface de Testes – Esses testes checam os métodos que fazem parte das interfaces definidas pela especificação.
- Teste de Stress – Para algumas especificações o teste de stress é incluído.

A página de diálogo que corresponde ao *Compliance Test Tool* Servidor pode ser visto na Figura 4:

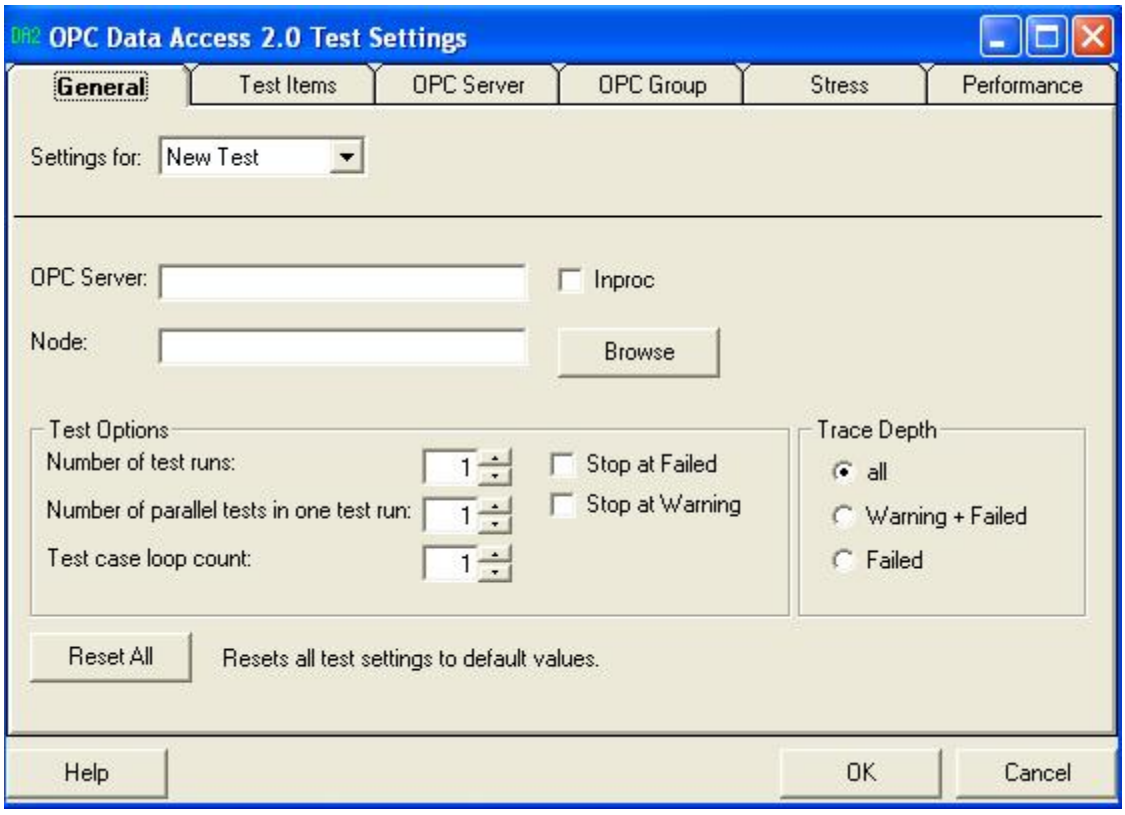


Figura 4 - Página de Configuração para o DA Compliance Test Tool

Fonte: OPC FOUNDATION, 2011

Como primeiro passo é necessário selecionar um Servidor DA, este pode estar localizado em um computador local ou remoto. O servidor selecionado é usado nos testes de interface e stress.

A aba "*Test Items*" habilita a definição de identificar um *namespace* para cada objeto *OPCItem* que é adicionado durante o teste.

Sob a aba "*OPCServer*" estão os itens que são selecionados para os testes da interface *IOPCItemMgt*.

Sob a aba "*OPCGroup*", as propriedades do objeto *OPCGroup* que serão criadas e o monitoramento de quantidade de vezes para realização dos testes são definidas.

A aba “*Stress*” permite o usuário estabelecer valores padrões para o teste de stress. Esta aba permite determinar quantos objetos *OPCServers* podem ser criadas no componente para ser testado.

Os resultados são gravados em um arquivo que pode ser emitido como um sumário.

A Figura 5 mostra as informações fornecidas pelo CTT quando finalizado os testes.

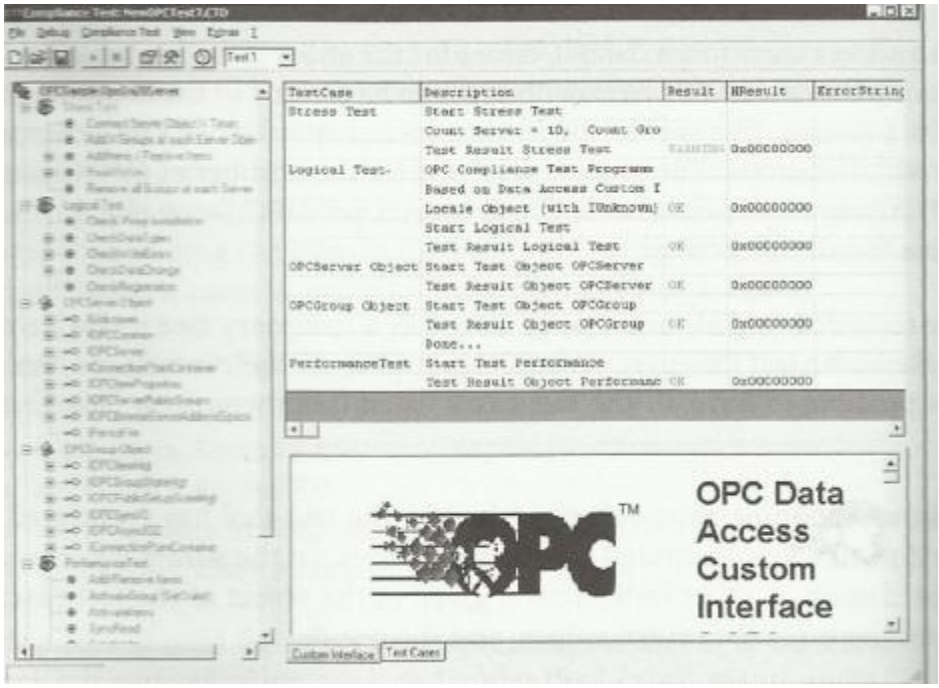


Figura 5 - Sumário de resultados de teste servidores DA

Fonte: BURKE;IWANITZ;LANGE,2010

3. OPC UA

3.1. Motivação OPC UA

Com a adoção do OPC em milhares de produtos, ele é utilizado atualmente como uma interface padrão entre sistemas de automação em diferentes níveis da pirâmide de automação.

Segundo (MAHNKE;LEITNER;DAMM,2009), o *OPC Unified Architecture* nasceu do desejo de se criar um verdadeiro substituto para tudo que era baseado nas especificações COM, sem perder alguma característica ou desempenho.

Adicionalmente o OPC deve abranger todos os requisitos para um sistema de interfaces para plataforma independente, com uma rica e extensa capacidade de modelamento, sendo hábil para descrever sistemas complexos. Uma larga série de aplicações onde o OPC é usado requer escalabilidade, partindo de sistemas embarcados através de SCADA e DCS até sistemas MES e ERP. A tabela 1 resume os principais requisitos para o OPC UA.

Tabela 1 - Requisitos para o OPC UA

Comunicação entre sistemas distribuidos	**Modelamento de dados**
Confiabilidade pela: • Robustez e tolerância a falhas • Redundância	Modelo comum para todos OPC Data
Plataforma Independente	Orientado à Objeto
Escalabilidade	Tipo de Sistemas Extensivos
Alta *performance*	Dados Complexos e métodos
Internet e Firewalls	Escalabilidade de um modelo simples

	até um complexo.
Segurança e controle de acesso	
Interoperabilidade	

Fonte: MAHNKE;LEITNER;DAMM,2009

Segundo, (BURKE;IWANITZ;LANGE,2010), algumas razões motivaram o desenvolvimento dessa nova geração, como:

- A descontinuação do COM/DCOM.
- Limitações do DCOM.
- A necessidade de utilização do OPC em plataformas Não-Windows.
- A necessidade de Alto Desempenho em comunicação via *Web Services*.
- A integração de todas as ferramentas em um modelo unificado.
- Suporte a estruturas complexas de dados
- Comunicação de dados de processo sem perda
- Aumento da proteção contra acessos não autorizados

3.2. OPC UA – Visão Geral

Para atingir os objetivos definidos, o *OPC Unified Architecture* construiu diversas camadas, como mostra a Figura 6:

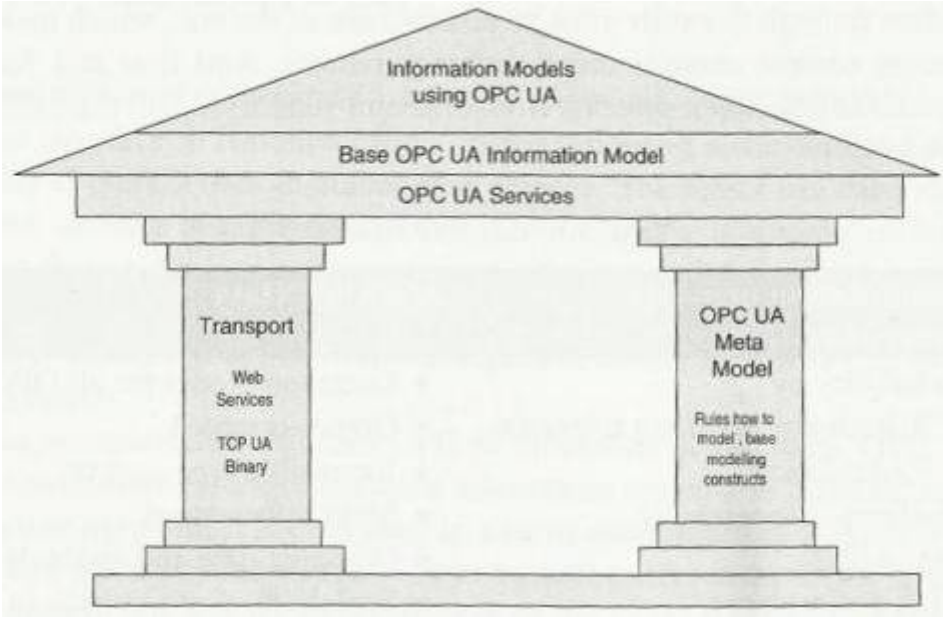


Figura 6 - A Fundação do OPC UA

Fonte: MAHNKE;LEITNER;DAMM,2009

Os alicerces do OPC UA são o mecanismo de transporte e modelamento de dados.

O transporte define diferentes mecanismos otimizados para diferentes casos. A primeira versão do OPC UA possui como definição a utilização do protocolo TCP para alto desempenho em comunicação intranet tão bem quanto um mapeamento para aceitar padrões internet, como *Web Services*, XML e HTTP (*OPC UA PART 1 - OVERVIEW AND CONCEPTS 1.01 SPECIFICATION*, 2009).

A modelagem de dados define as regras e a base de blocos de construção necessários para expor um modelo de informação com o OPC UA, além determinar os pontos de entrada no espaço de endereços e tipos de bases usadas para construir uma hierarquia.

Os Serviços UA são a interface entre servidores, como fornecedores de informação, e clientes, como consumidores desta informação. Eles são usados no mecanismo para troca de dados entre cliente e servidor.

Este conceito básico do OPC UA possibilita que um Cliente OPC UA acesse as menores partes possíveis dos dados, sem a necessidade de entender integralmente o modelo exposto por um sistema complexo.

A figura 7 apresenta as diferentes camadas dos modelos de informação definidas pela OPC.

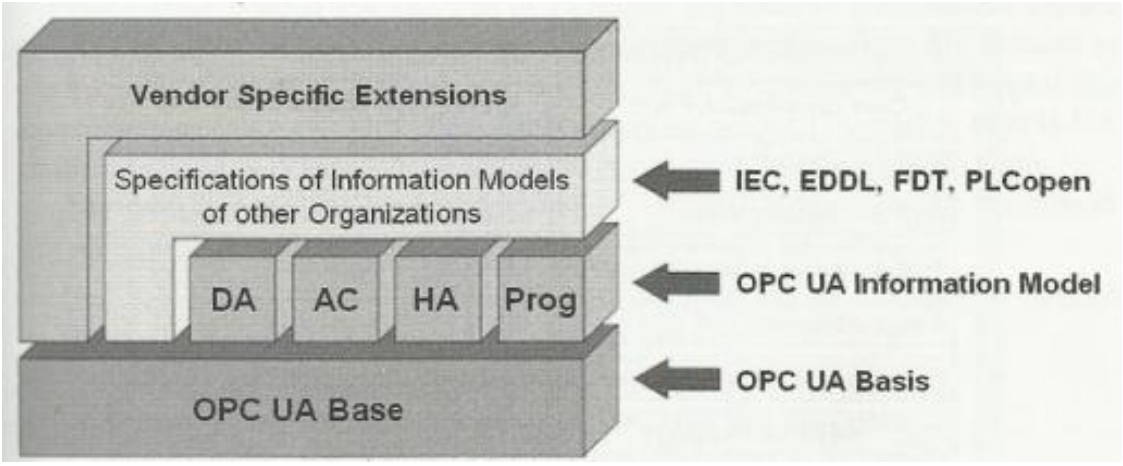


Figura 7 - Camadas Arquitetura OPC UA

Fonte: MAHNKE;LEITNER;DAMM,2009

Para cobrir com sucesso todas as características conhecidas no OPC Clássico, modelos de informação para as informações do processo foram definidas em cima da base do OPC UA.

Segundo OPC UA PART 1 - OVERVIEW AND CONCEPTS 1.01 SPECIFICATION (2009), os modelos de informação são definidos como:

- O DA(*Data Access*) define os dados específicos de automação, tais como modelagem de dados analógicos ou discretos e a qualidade do serviço. Todas outras características do DA são cobertas pela OPC UA base.
- *Alarm & Conditions* (AC) específica um modelo avançado de gerenciamento de alarmes e condições de monitoramento.

- *Historical Access* (HA) define o mecanismo para acessar dados históricos e eventos. *Programs* (Prog) específica um mecanismo para iniciar, manipular e monitorar a execução de programas.

Outras organizações construíram seus modelos em cima da base UA ou do modelo de informação OPC, expondo suas específicas informações via OPC UA. Alguns exemplos podem ser citados, como: o *Field Device Integration* (FDI) combinado ao *Eletronic Device Description Language* (EDDL), e o *Field Device Tool* (FDT), ambos usados para descrever, configurar e monitorar equipamentos, e o *CLPopen*, como um padrão para linguagens de programação para PLC (MAHNKE;LEITNER;DAMM,2009).

3.3. OPC UA Especificações

As especificações do OPC UA são divididas em diferentes partes, conforme exigido pelo padrão IEC. O OPC UA é conhecido como o padrão IEC 62541. A figura 8 mostra uma visão geral de todas as especificações, divididas em "*Core Specifications*" e "*OPC UA Information Models*", definindo respectivamente a base para o OPC UA e o tipo específico de acesso.

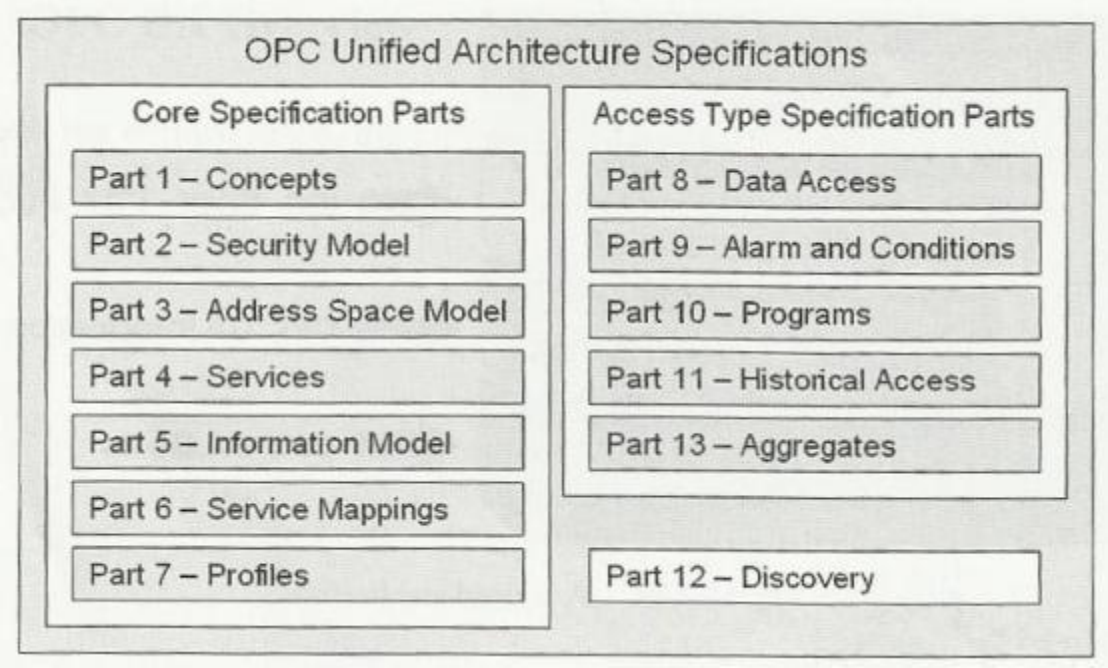


Figura 8 - Especificações OPC UA

Fonte: MAHNKE;LEITNER;DAMM,2009

Segundo (MAHNKE, WOLFGANG; 2009), as duas primeiras partes não são normativas. O "*UA Part 1*" corresponde ao conceito do OPC UA e o "*UA Part 2*" descreve as necessidades e os modelos de segurança do OPC UA.

As partes 3 e 4 são as mais importantes para projeto e desenvolvimento de aplicações UA, explicando como modelar e acessar as informações. O "*Address Space Model*" – "*UA Part 3*" específica as construções dos blocos com as instâncias e o tipo de informação para construir o "*OPC UA Server Address Space*".

Para o "*Abstract UA Services*" definido no "*UA Part 4*", representa as possíveis interações entre os Clientes UA e as aplicações dos Servidores UA. O cliente usa o serviço para encontrar e acessar as informações fornecidas pelo servidor. A Figura 9 apresenta as camadas de comunicação da arquitetura do OPC UA.

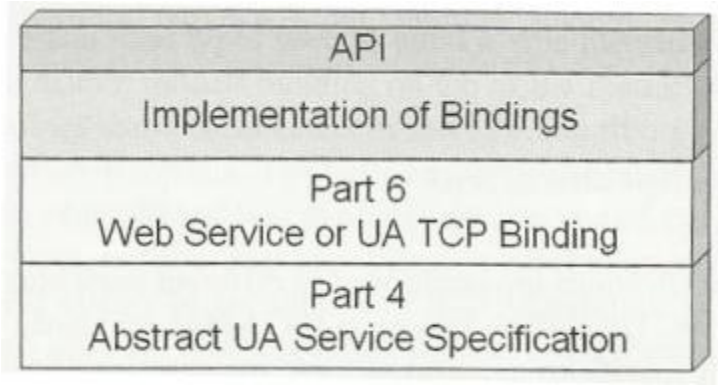


Figura 9 - Camadas Arquitetura de Comunicação do OPC UA

Fonte: MAHNKE;LEITNER;DAMM,2009

O Mapeamento do serviço de mensagens para o OPC UA, os mecanismo de segurança aplicados nas mensagens, e o confiável meio de transporte da mensagem são definidas no "*UA Part 6*". Apenas os implementadores dos "*UA Stacks*" necessitam entender completamente esta especificação.

A base do "*Information Model*" especificado no "*UA Part 5*" fornece o framework para todos "*Information Models*" usando OPC UA.

Os "*Profiles*" são definidos como úteis subconjuntos de características do OPC UA no "*UA Part 7*". Um subconjunto deve ser implantado completamente pela aplicação UA para garantir a interoperabilidade do subconjunto definido.

A Especificação define os subconjuntos em dois níveis. O primeiro nível são unidades de conformidade definindo um pequeno conjunto de funcionalidades que são sempre utilizadas juntas e podem ser testadas com o CTT e verificadas como unidade.

O segundo nível são perfis compostos por uma lista de unidades de conformidade. Um perfil deve ser implantado completamente e será verificado como um conjunto completo durante a certificação dos produtos OPC UA. A

lista de perfis suportados e usados é trocada durante o estabelecimento da conexão entre cliente e servidor, permitindo a aplicação determinar se as características necessárias são suportadas pelo parceiro de comunicação.

O modelo de informação DA define como representar e usar dados de automação e características especificas como unidade de engenharia no “*UA Part 8*”.

O modelo de informação AC especifica o processo de alarmes e condição de monitoramento de estados das máquinas e tios de eventos no “*UA Part 9*”.

O modelo de informação Prog define a base do estado da máquina para a execução, manipulação, e monitoramento de programas no “UA Part 10”.

O modelo de informação HA (*Historical Access*) no “*UA Part 11*” especifica o uso do histórico de serviços de acesso e como apresentar informações sobre a configuração dos dados e histórico de eventos.

Os valores agregados de dados amostrados são especificados no “*UA Part 13*” e “*UA Part 12*”, definindo como encontrar servidores na rede e como um cliente pode obter a informação necessária para habilitar e estabelecer a comunicação para um certo servidor.

3.4. OPC UA Software Layers

O OPC UA usa o conceito similar de cliente-servidor como o OPC Clássico. Uma aplicação que quer expor sua própria informação para outras aplicações é chamada de Servidor UA e uma aplicação que quer consumir informação de outra aplicação são chamadas de Cliente UA.

Uma típica aplicação OPC UA é composta de três camadas de software apresentadas na Figura 10.

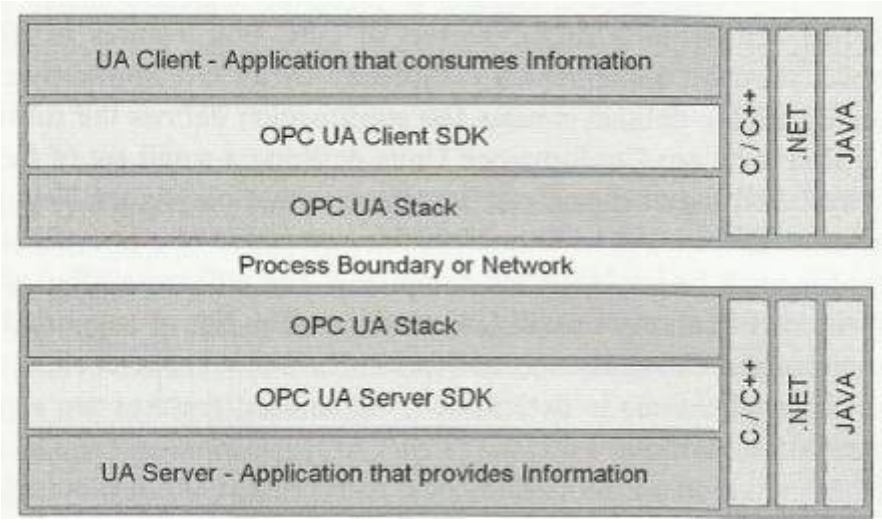


Figura 10 - OPC UA Camadas de Software

Fonte: MAHNKE;LEITNER;DAMM,2009

O software completo pode ser desenvolvido com C/C++, NET, ou JAVA. O OPC UA não é limitado nestas linguagens de programação e plataformas de desenvolvimento, mas apenas estes ambientes são usualmente utilizados para desenvolvimento do *OPC Foundation UA*.

Uma aplicação é um sistema que quer fornecer ou consumir dados via OPC UA. Contém a funcionalidade específica para a aplicação e o mapeamento para o OPC UA, utilizando um *OPC Stack* e um *OPC UA Software Developing Kit* (SDK).

Um Cliente ou Servidor UA SDK desenvolve funcionalidades comuns que estão na parte da camada de aplicação, assim *UA Stacks* implementa apenas o canais de comunicação. Um *OPC UA SDK* reduz o esforço e facilita o desenvolvimento da interoperabilidade para uma aplicação do OPC UA.

Um *OPC UA Stack* implanta os “*Service Mappings*” definidos no “*UA Part 6*”. Uma “*Stack*” é usada para realizar Serviços UA em limites de processo ou de rede. O OPC UA define três camadas de “*Stack*” e diferentes perfis para cada camada.

A camada de codificação da mensagem (*Message Serialization*) define a serialização dos parâmetros de Serviço em formato binário ou XML. A camada de Segurança da mensagem (*Messsage Security*) define como as mensagens devem estar seguras usando o padrão de segurança *Web Service* ou uma versão binária do UA padrão *Web Service*.

A camada de transporte (*Message Transport*) define o protocolo de rede usado, que pode ser *UA TCP* ou HTTP e SOAP para *Web Serviços*. (MAHNKE;LEITNER;DAMM,2009).

A Figura 11 ilustra as camadas de comunicação Stack. O desenvolvimento das camadas em *UA Stack* e o API resultante para aplicação não faz parte da especificação OPC UA.

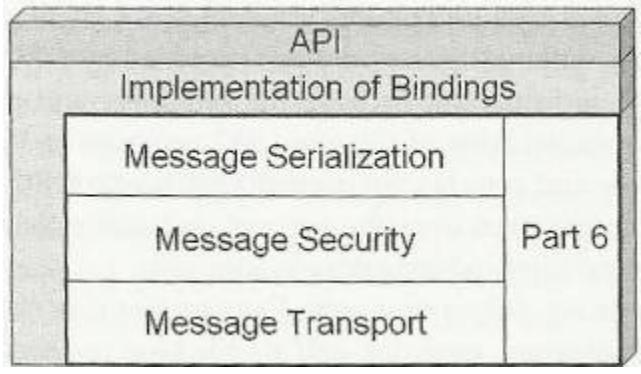


Figura 11 - Camadas de Comunicação UA Stack definidas no UA Part 6

Fonte: MAHNKE;LEITNER;DAMM,2009

Com as implementações em ANSI C/C#, NET, e JAVA , os principais ambientes de desenvolvimento e linguagens de programação são cobertas pelos *UA Stacks* desenvolvidos e mantidos pela *OPC Foundation.*

3.5. Compliance Test Tool OPC UA

O OPC UA CTT fornece perfis de testes com base em uma ferramenta de teste em uma plataforma indepedente (BURKE;IWANITZ;LANGE,2010). O OPC UA CTT suporta a geração de resultado de certificação, e também oferece suporte a depuração pelo desenvolvedor. Essa simples ferramenta pode operar em modo Cliente ou Servidor.

3.5.1. Servidor OPC UA CTT

Quando o projeto de um servidor é selecionado, o usuário deve fornecer algumas informações de configuração a respeito do servidor, como a URL do Servidor. Uma vez iniciadas as ferramentas de testes, o usuário é apresentado a uma lista de Perfis. Por padrão os Perfis expostos pelo servidor são checados e a ferramenta pode adicionar perfis adicionais para a lista de testes.

Quando a ferramenta está operando em modo depuração, testes individuais podem ser selecionados e executados. A ferramenta também permite a realização do teste passo a passo, permitindo a depuração das falhas conforme elas ocorrem.

Com a operação em modo certificação todos os perfis configurados serão automaticamente testados (Figura 12).

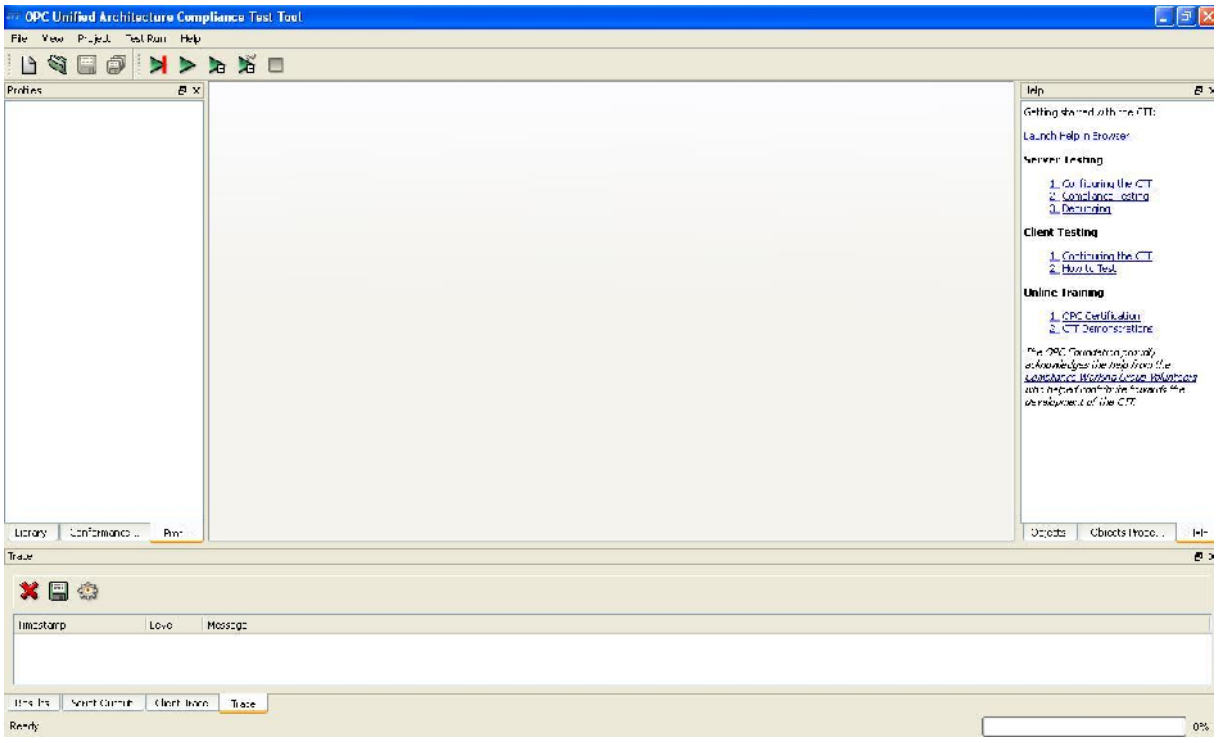


Figura 12 - Servidor OPC UA CTT

Fonte: BURKE;IWANITZ;LANGE,2010

4. Migração OPC UA

4.1. Visão Geral

OPC UA fornece estratégias de migração para diferentes necessidades e níveis de adoção do OPC UA. O primeiro nível não requer mudanças nos produtos existentes. *Wrappers* and *Proxies* fornecidos pela *OPC Foundation* estão disponíveis para traduzir diferentes interfaces do OPC Clássico para o OPC UA e vice-versa. Este nível é apropriado para integração com a base instalada de Produtos OPC.

Para Vendedores de Produtos OPC, os outros níveis de migração são mais importantes. O segundo nível usa os ”*mappings*”, já descritos na seção 3.4, para exibir algumas características dos produtos existentes com OPC UA. Isto não requer qualquer mudança sobre as interfaces internas para informação de acesso a um sistema apresentado como OPC Clássico hoje.

A vantagem sobre os *Wrappers* é a mais alta performance, poucos limitações, e menos esforços de manutenção por evitar mais uma camada adicional de software para o *Wrapper*. Do ponto de vista do produto, o esforço não é muito mais do que utilizando a integração de *wrappers*, considerando apenas o trabalho de desenvolvimento.

4.2. Wrappers

O *OPC UA Wrapper* é usado para permitir o Cliente OPC UA acessar Servidores do OPC Clássico. Assim um componente wrapper é um Cliente OPC para um padrão OPC Clássico acessando um servidor e ao mesmo

tempo o wrapper é um Servidor OPC UA permitindo Clientes UA conversem com o servidor wrapper.

A Figura 13 mostra os componentes de um wrapper UA fornecendo acesso para um Servidor OPC DA para clientes UA.

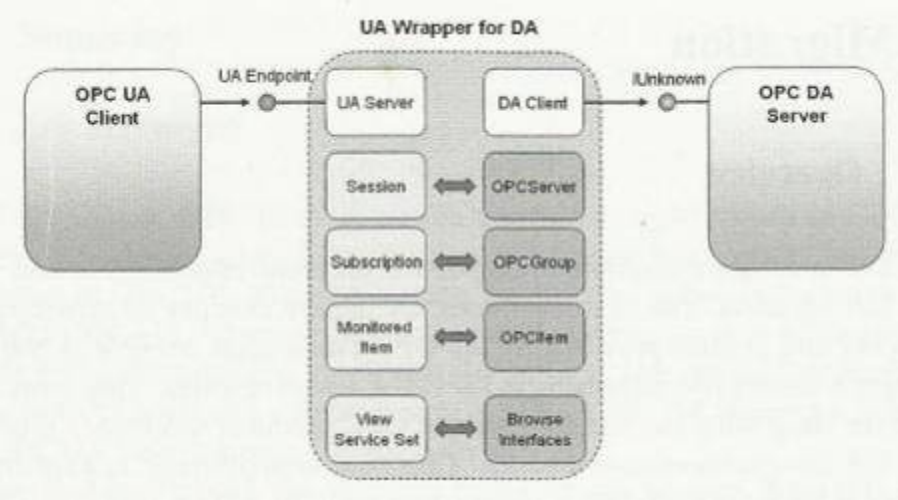


Figura 13 - Wrapper UA fornencendo acesso ao Servidor OPC DA

Fonte: MAHNKE;LEITNER;DAMM,2009

O *UA DA wrapper* desenvolve um servidor UA exibindo "*UA Endpoints*", permitindo clientes UA acessar os dados de qualquer fornecedor em uma interface conforme do OPC DA. Uma sessão criada por um cliente UA resulta na criação de um objeto *OPCServer* no servidor OPC DA e uma subscrição items monitorados são mapeados para um *OPCGroup* com *OPCItems*.

4.3. Proxies

OPC UA Proxies são usados para permitir clientes OPC Clássico acessar Servidores OPC UA. Assim um componente Proxy é um cliente OPC UA acessando um Servidor UA e ao mesmo tempo o *Proxy* é o Servidor

clássico OPC exposto com uma interface COM, permitindo clientes usar um padrão do OPC Clássico para acessar Servidores OPC UA.

A Figura 14 apresenta os componentes de um Proxy fornecendo acesso de um Servidor OPC UA para um cliente OPC DA.

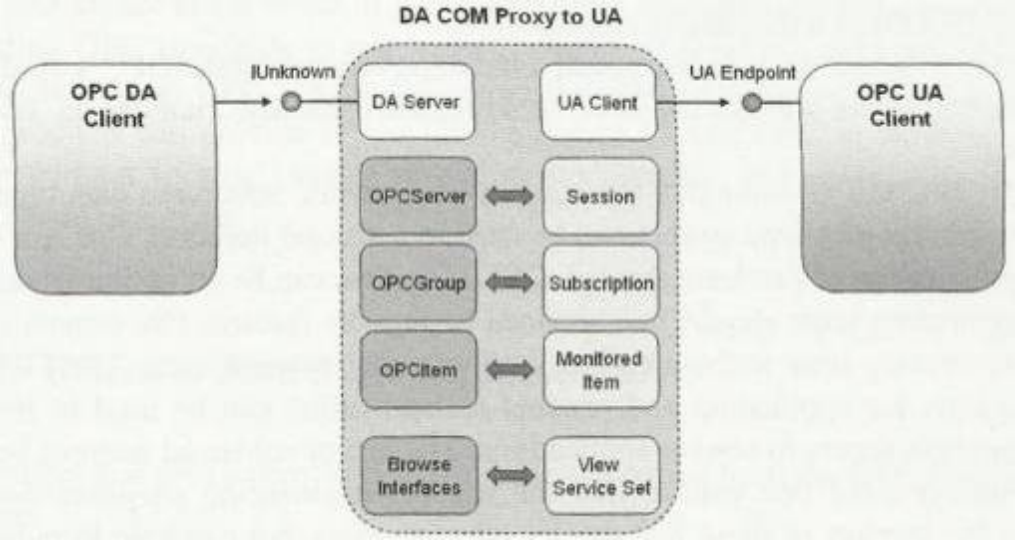


Figura 14 - *Proxy* DA COM fornecendo acesso para um Servidor OPC UA

Fonte: MAHNKE;LEITNER;DAMM,2009

5. Sistemas Supervisórios

5.1. Introdução

Segundo (MORAES;CASTRUCCI,2007), Sistemas Supervisórios são sistemas digitais de monitoração e operação da planta que gerenciam variáveis de processo. Podem ser caracterizados e classificados pela complexidade, robustez e número de pontos (Entrada e Saídas) monitorados. São divididos basicamente em IHM/HMI(Interface Homem Máquina / *Human Machine Interface*) e SCADA(*Supervisor Control and Data Acquisition*). A IHM é situada normalmente próxima à linha de produção, instalada nas máquinas, traduzindo os sinais vindos dos PLCs para sinais gráficos (Figura 15).

Figura 15 - Exemplo de IHM com visão de Processo

Fonte: DIYTRADE,2011

O SCADA foi criado para supervisão e controle de quantidades elevadas de variáveis de entrada e saídas digitais e analógicas distribuídas (Figura 16).

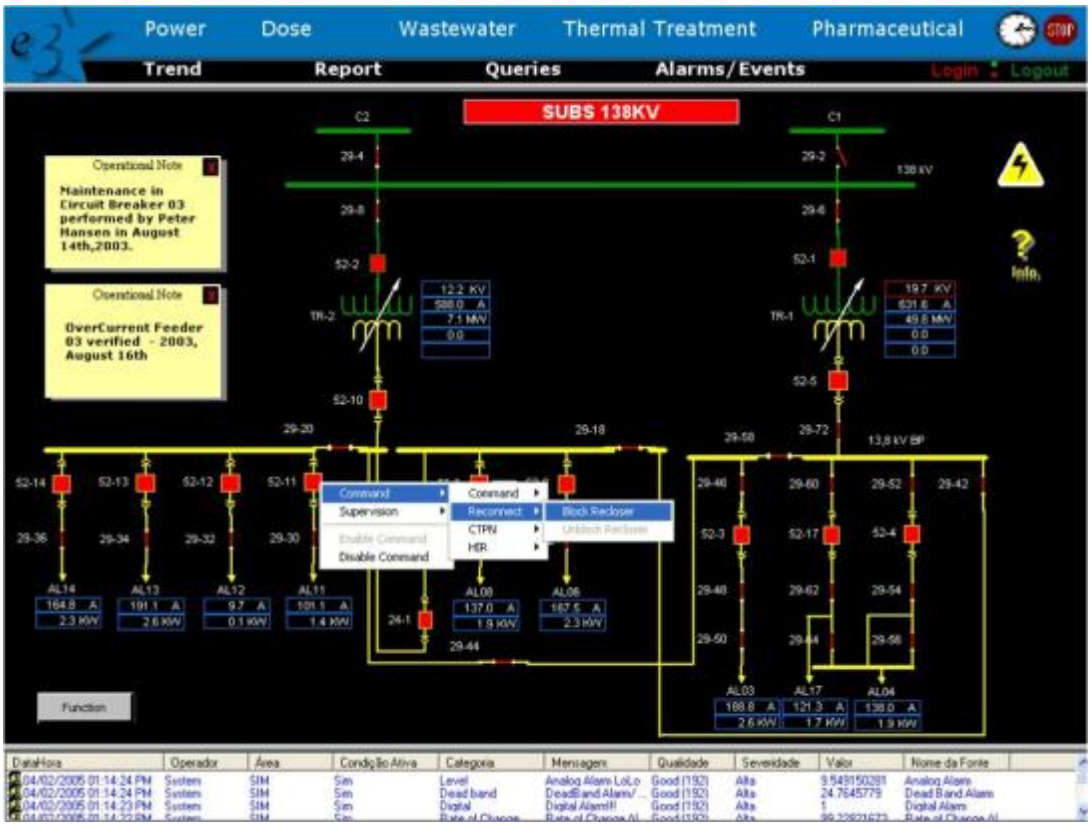


Figura 16 - Exemplo de uma tela de supervisório para controle e monitoração

Fonte: LIMA,2011

5.2. SCADA

Segundo (ALBUQUERGUE,2007), o sistema SCADA é responsável pela coleta e transferência de informações lógicas e analógicas sobre o estado do sistema, por sua exibição na sala de controle e pelo comando remoto de dispositivos.

O SCADA coleta as informações de processo através de uma RTU (*Remote Terminal Unit*), transferindo-o de volta para o sistema central, realizando a análise necessária e controle e, em seguida, apresentando a informação sobre uma série de telas para a operação.

As ações de controle necessárias são levadas de volta para o processo. Embora inicialmente a RTU seja muitas vezes um dispositivo dedicado, CLPs são usados frequentemente como RTUs atuais. CLPs e/ou DCS (*Distributed*

Control Systems) são utilizados como mostra Figura 17.

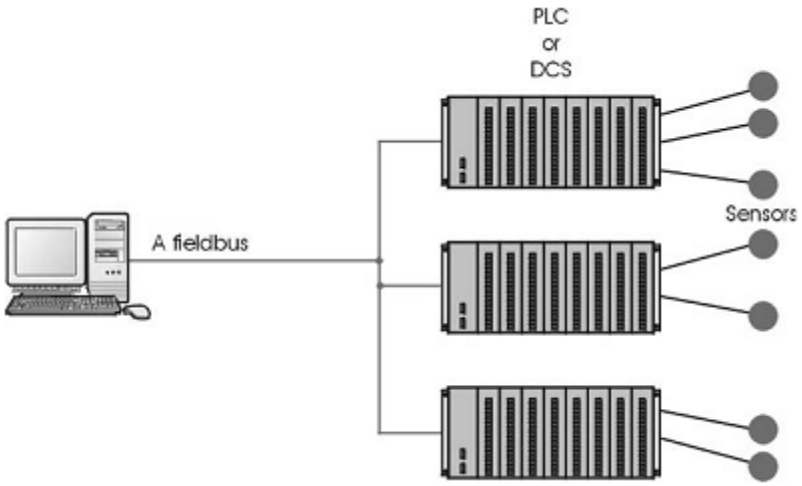


Figura 17 - PC para CLP ou DCS com um Barramento de Comunicação e sensores

Fonte: BAILEY;WRIGHT,2003

Com a exigência crescente de sistemas menores e mais inteligentes, atualmente alguns sensores estão sendo projetados com a inteligência de CLPs e DCSs. Esses dispositivos são conhecidos como IEDs (*Intelligent Electronic Devices*). Os IEDs são conectados em um fieldbus, tais como Profibus, DeviceNet ou Fieldbus Foundation para o PC (Figura 18).

Segundo (CLARKE;REYNDERS;WRIGHT,2004) estes sensores incluem a inteligência suficiente para adquirir dados, comunicar com outros dispositivos e realizar controle. Normalmente, um IED pode combinar um sensor de entrada analógica, saída analógica, controle PID, sistema de comunicação e memória de programa em um único dispositivo.

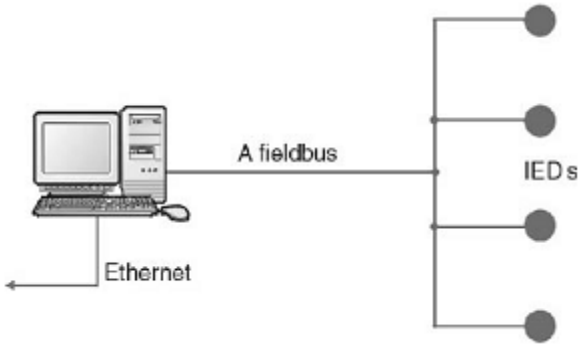


Figura 18 - PC para IED usando barramento de comunicação

Fonte: BAILEY;WRIGHT,2003

A comunicação de qualquer sistema SCADA, que é a sua principal funcionalidade, ocorre principalmente com:

- Equipamentos de campo
- PLCs/RTUs
- Outras estações SCADA
- Outros sistemas

Segundo (SILVA;SALVADOR, 2010) os sistemas SCADA geralmente dividem suas principais tarefas em blocos ou módulos, que vão permitir maior ou menor flexibilidade e robustez, de acordo com a solução desejada. De forma geral, essas tarefas podem se divididos em:

- Núcleo de processamento.
- Comunicação com PLCs/RTUs.
- Gerenciamento de Alarmes.
- Históricos e Banco de Dados.
- Lógicas de programação interna (*Scripts*) ou controle.
- Interface gráfica.

- Relatórios.
- Comunicação com outras estações SCADA.
- Comunicação com Sistemas Externos / Corporativos.
- Outros.

5.2.1. Limites dos sistemas SCADA

As diversas operações realizadas pelos sistemas SCADA causam a necessidade de agregação de recursos aos sistemas computacionais.

Segundo NASCIMENTO FILHO (2005), um dos grandes problemas de relação de custo x benefício das soluções para supervisão de sistemas industriais é a utilização de *drivers* específicos e proprietários para comunicação, pois aumenta o custo de desenvolvimento(Software e Hardware) e dificulta futuras ampliações do sistema SCADA.

Essa é uma das situações mais críticas combatidas pelos sistemas SCADA e que motivaram o desenvolvimento de padrões de interoperabilidade (OPC COMMON DEFINITIONS AND INTERFACES, 1998).

5.3. Elipse E3

Dentre os Sistemas SCADA, o Elipse E3 foi utilizado no trabalho por suportar as tecnologias COM, DCOM, ActiveX e OPC,e possuir recursos que facilitam a configuração de servidores OPC, além de ser uma ferramenta que possui uma grande quantidade de informações (Manuais e Tutoriais) disponíveis gratuitamente, auxiliando o rápido entedimento da ferramenta.

De acordo com KICHALOWSKY(2010), o E3 é a terceira geração de sistemas HMI/SCADA da Elipse Software e representa a evolução dos

tradicionais sistemas cliente/servidor (de duas camadas) para um modelo de múltiplas camadas (composto de servidores, regras de aplicação ou de negócio e estações clientes). O E3 possui interoperabilidade com centenas de dispositivos através da utilização de *drivers* proprietários, desenvolvidos pela própria Elipse Software, e OPC (Figura 19).

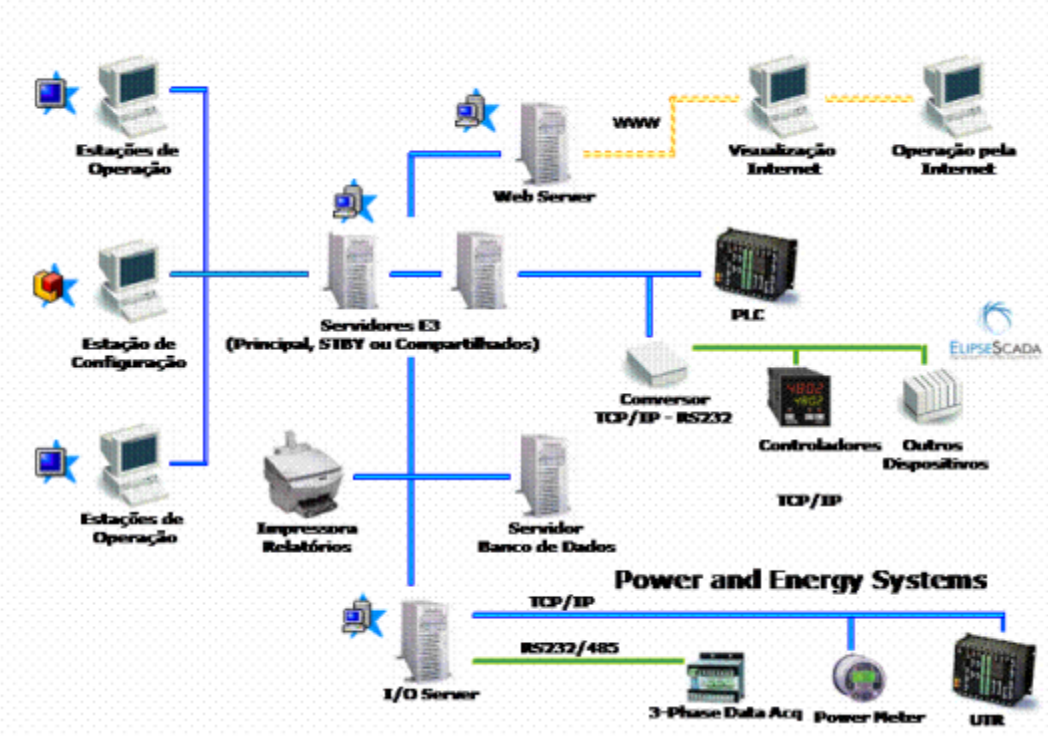


Figura 19 - Exemplo de arquitetura de uma aplicação E3

Fonte: KICHALOWSKY,2010

O E3 é composto por três aplicativos principais: E3Server, E3Viewer e E3Studio. Neste trabalho vamos dar um enfoque principalmente no E3 Server.

5.4. E3 Server

Segundo (KICHALOWSKY,2010) e (ALBUQUERGUE,2007) o E3 Server é um servidor de aplicações, onde são gerenciados os processos de execução do software e processada a comunicação entre eles. Dentre as ações operadas por este componente estão:

- Envio das informações gráficas e dados para os clientes.
- Gerência dos processos de E/S, comunicação com os diversos pontos de aquisição.
- Controle de licenças.
- Cliente e servidor OPC.
- Sincronismo de alarmes e bases de dados.

5.5. Elipse Plant Manager

O Elipse Plant Manager (EPM) é um software desenvolvido com a tecnologia OPC UA, e por esse motivo foi escolhido para realizar os testes com o Wrapper OPC, permitndo o Cliente OPC UA(EPM) acessar Servidores do OPC Clássico(E3).

De acordo com a ELIPSE SOFTWARE (2011), o EPM é uma ferramenta que atua na coleta de informações de tempo real ou históricas, como também em sua contextualização, compactação, disponibilização e análise, provendo uma camada de dados entre o universo de tempo-real e o mundo orientado a transações de negócios, finanças e sistemas no campo corporativo.

Com o EPM, os dados obtidos de várias fontes como controladores de processo, medidores, SCADA’s e SDCD´s, podem ser armazenados, por longos períodos de tempo, de forma compacta, centralizada, segura e tolerante a interrupções na comunicação, possibilitando que sejam consultados de forma rápida por diversos tipos de aplicações.

6. REDES INDUSTRIAIS

Segundo (FRANCO;LENARTH,2010), por definição, uma rede industrial (Figura 20) é um sistema de comunicação bi-direcional em tempo real que permite a troca de informação digital entre os dispositivos de nível de campo e os dispositivos de monitoramento e controle.

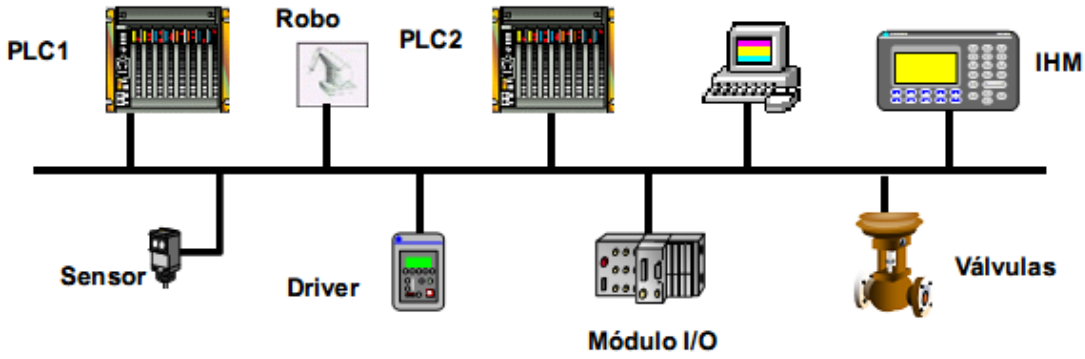


Figura 20 - Exemplo de Rede Industrial

Fonte: FRANCO;LENARTH,2010

Essas redes são mais robustas se comparadas com redes de computadores, tanto com relação ao cabeamento, devido à necessidade de se ter uma maior proteção a ruídos, quanto conectores mais forte e resistente.

Em sua maioria, as redes industriais são determinísticas, ou seja, realizam a comunicação em tempos pré-estabelecidos, e requerem as respostas e atualizações em tempo real.

Um exemplo de rede industrial é protocolo Modbus, que será descrito no próximo sub item.

6.1. MODBUS

O *Modbus* foi originalmente desenvolvido pela *Modicon*, e hoje, é gerenciado pela *Modbus-IDA User Organization*. É um protocolo aberto, baseado no modelo Mestre/Escravo. Neste modelo os escravos não podem dialogar entre si, e todo controle e informação deverão ocorrer no mestre, sendo possível o mestre requisitar apenas uma mensagem particular a algum escravo ou para todos os escravos da rede.

O *Modbus* pode ser utilizado com diversas camadas físicas, e está situado na camada de apresentação e aplicação do Nível do modelo OSI. Ele permite uma comunicação cliente/servidor de equipamentos conectados entre diferentes tipos de barramentos ou redes.

O *Modbus-TCP* siginifica que o protocol Modbus é usado sobre a camada Ethernet-TCP/IP (Figura 21). O *Modbus-TCP* é uma Rede Industrial Ethernet aberta, que foi especificada pela *Modbus-IDA User Organization* com a cooperação da *Internet Engeneering Task Force* (IETF).

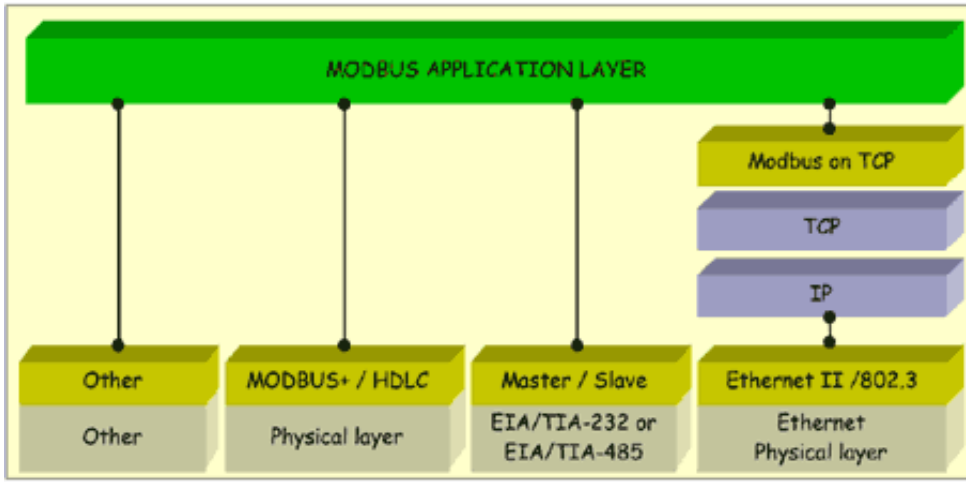


Figura 21 – Modbus sobre TCP/IP

Fonte: HMS ANYBUS,2011

O *Modbus* possui uma gama de produtos adicionais, que agora

consistem no clássico *Modbus-RTU* (via RS-232 or RS-485), *Modbus-Plus* (Comunicação Alta Velocidade via *Token Passing*) e o *Modbus-TCP* (*Ethernet-TCP/IP*- baseado na comunicação cliente/servidor). Todas estas versões compartilham o mesmo protocolo de aplicação, que especifica um módulo universal para dados do usuário e serviços de comunicação.

7. DESENVOLVIMENTO SERVIDOR OPC DA E UA

7.1. Ambiente de desenvolvimento

Para desenvolver os servidores OPC DA e UA foi necessário o uso de ferramentas de programação de alto nível. No caso do desenvolvimento das classes modularizadas nos subsistemas de comunicação e tratamento de protocolo, assim como no acesso ao banco de dados foi utilizado o MS Visual Studio. Essa ferramenta oferece possibilidade de programação na linguagem C#, essencial para o desenvolvimento de cada módulo utilizado nos servidores, além de se integrar com a ferramenta automatizada de geração de interfaces OPC. O banco de dados utilizado para o armazenamento e a definição do Address Space foi o MySQL 5.0.

7.2. Toolkit de criação de interfaces

Sistemas que seguem o padrão OPC podem ser criados tanto diretamente por programadores experientes nas tecnologias adotadas por esses padrões quanto através de ferramentas que automatizam parte da codificação.

Vários *toolkits* foram utilizados, sem perda da generalização dos passos definidos para o desenvolvimento de servidores OPC, dentre os quais o da empresa *Softing*, da *Northern Dynamic (SLIK-DA)* e da própria *OPC Foundation*.

Assim como os demais, esses toolkits devem ser instalados sobre o sistema operacional *Microsoft Windows*, onde já se encontre instalado

anteriormente o *MS Visual Studio*. Após sua instalação, é adicionada a opção de criação de servidores OPC entre as soluções possíveis para desenvolvimento. Caso essa opção seja escolhida, o *toolkit* apresenta um *wizard* para a definição de algumas características relevantes na implementação de um servidor OPC. Essas telas requerem que algumas decisões sejam tomadas sobre o tipo de servidor OPC que se deseja desenvolver. Todo o código de interfaces é gerado automaticamente, com seus métodos vazios. Dessa forma, pode-se partir para o passo de "escolha de componentes existentes" com o foco na remoção daqueles que não se mostrarem necessários.

A seguir será apresentado um passo a passo para o desenvolvimento de Servidores OPC baseados em Toolkits, o método apresentado foi utilizado para o desenvolvimento do servidor com o toolkit da empresa *MatrikonOPC(SLIK-DA), porém este método pode ser estendido aos toolkits da empresa Softing OPC e da própria OPC Foundation.*

Para o desenvolvimento do OPC Server, após instalação do toolkit deve-se iniciar um novo Projeto no Visual Studio.

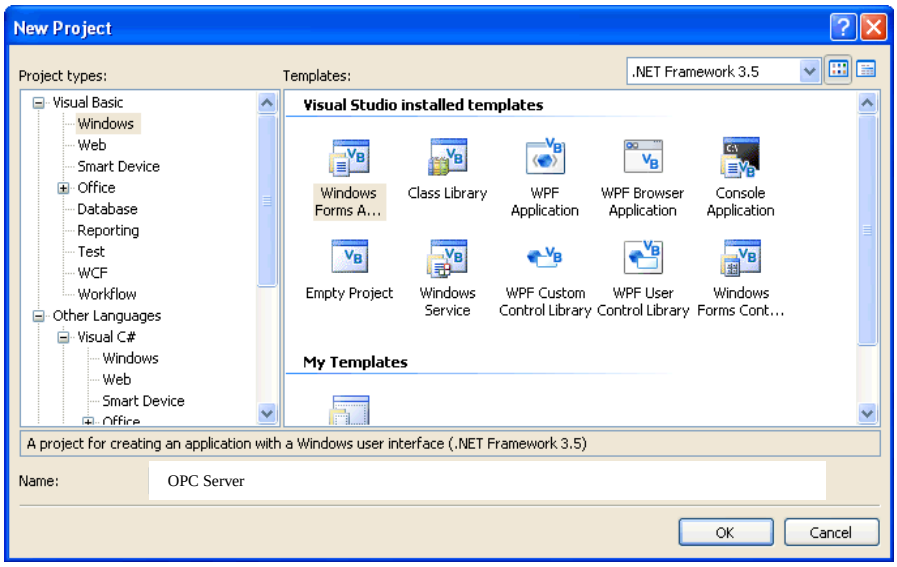


Figura 22 – Criação de um novo Projeto

Fonte: SLIKDA Tutorial

Com o novo projeto aberto deve-se inserir o toolkit do OPC Server ao programa, para isso o desenvolvedor selecionará o local onde gostaria de adicionar o componente(toolkit) e criar a sua Categoria, por exemplo “OPC Server”. Posteriormente é necessário selecionar os itens que integram ferramenta (Figura 25).

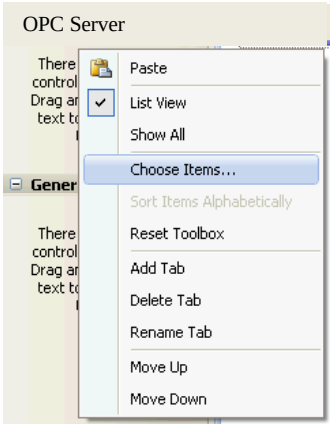


Figura 23 – Seleção de Itens para o toolbox

Fonte: SLIKDA Tutorial

Para seleção dos itens, abra a pasta onde o toolkit foi instalado, direcione-se a pasta “Redistributables” e selecione o driver ligado a

interoperabilidade do servidor, neste caso o arquivo Interop.NDISLIKDA.dll (Figura 24). Se houver alguma dúvida com relação à escolha do driver correto, verifique o tutorial/manual do desenvolvedor do toolkit.

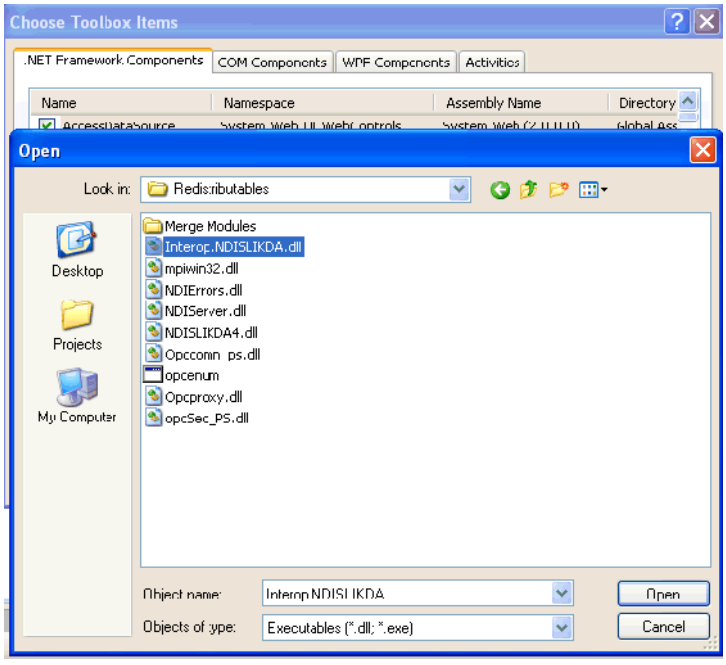


Figura 24 – Seleção do driver

Fonte: SLIKDA Tutorial

Posteriormente, selecione o toolkit instalado para compor o toolbox do Visual Studio(Figura 25).

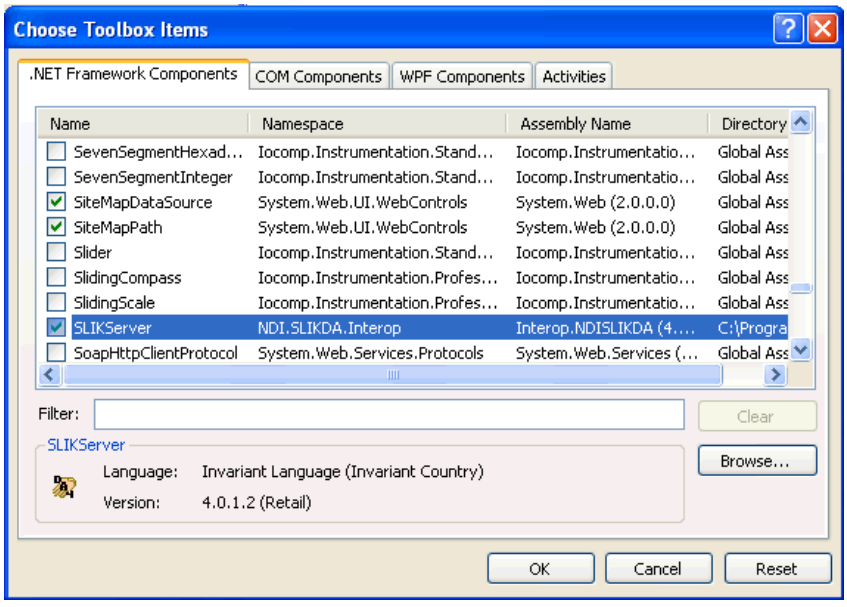


Figura 25 – Seleção do driver – Toolbox Items

Fonte: SLIKDA Tutorial

Após a correta inserção do toolkit como um componente/toolbox do Visual Studio deve-se abrir uma “Form” do programa e adicionar o ícone do referente ao toolkit no novo projeto.

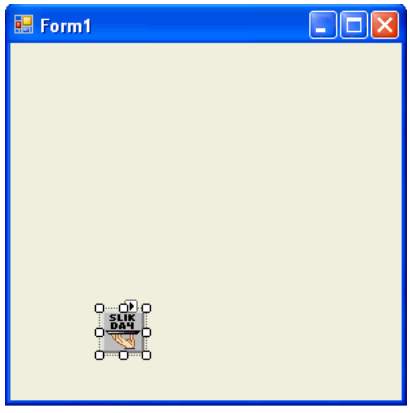


Figura 26 – Inserção do Icone “SLIK-DA” na “Form”

Fonte: SLIKDA Tutorial

Neste momento é necessário ajustar as propriedades do toolkit, identificando OPC Server através do “Class ID” (CLSID), que por sua vez é um nome alfanumérico único que identificara o OPC Server. Para obter esta identificação é necessário apenas localizar a propriedade “CLSID” e escolher “AutoGenerate”.

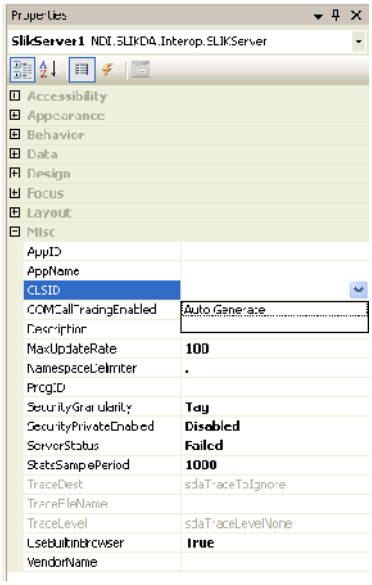


Figura 27 – Identificação do CLSID

Fonte: SLIKDA Tutorial

Após selecionar “AutoGenerate” será gerado um número alfanumérico único para identificação do servidor. O mesmo procedimento deverá ser executado para criar uma identificação para Aplicação, o “Application ID - AppID”.

Para finalizar a identificação do OPC Server é necessário abrir o código fonte “Form” e definir a identificação de algumas propriedades, como as citadas abaixo:

- ProgID = Este é o nome do servidor que os usuários verão quando se navega uma lista de servidores disponíveis OPC em seu computador.
- AppName = Este é o nome do OPC Server
- Description = Descrição básica do Servidor
- VendorName = Nome do desenvolvedor ou companhia

Posteriormente é necessário registrar o Servidor no Windows, justamente para ele ser “visto” pelos Clientes OPC, fazendo isso através de um

parâmetro de linha de comando.

Esse método de registro é escolhido para evitar o registro/cancelamento do servidor OPC cada vez que o software é aberto ou fechado. Isso impede que o servidor seja usado por clientes OPC enquanto o servidor não está funcionando. Portanto, é necessário registrar o servidor através de um parâmetro de linha de comando uma vez. Da mesma forma, o servidor pode cancelar seu registro através do parâmetro de linha de comando.

Um exemplo da linha de comando utilizada é apresentado na figura 28.

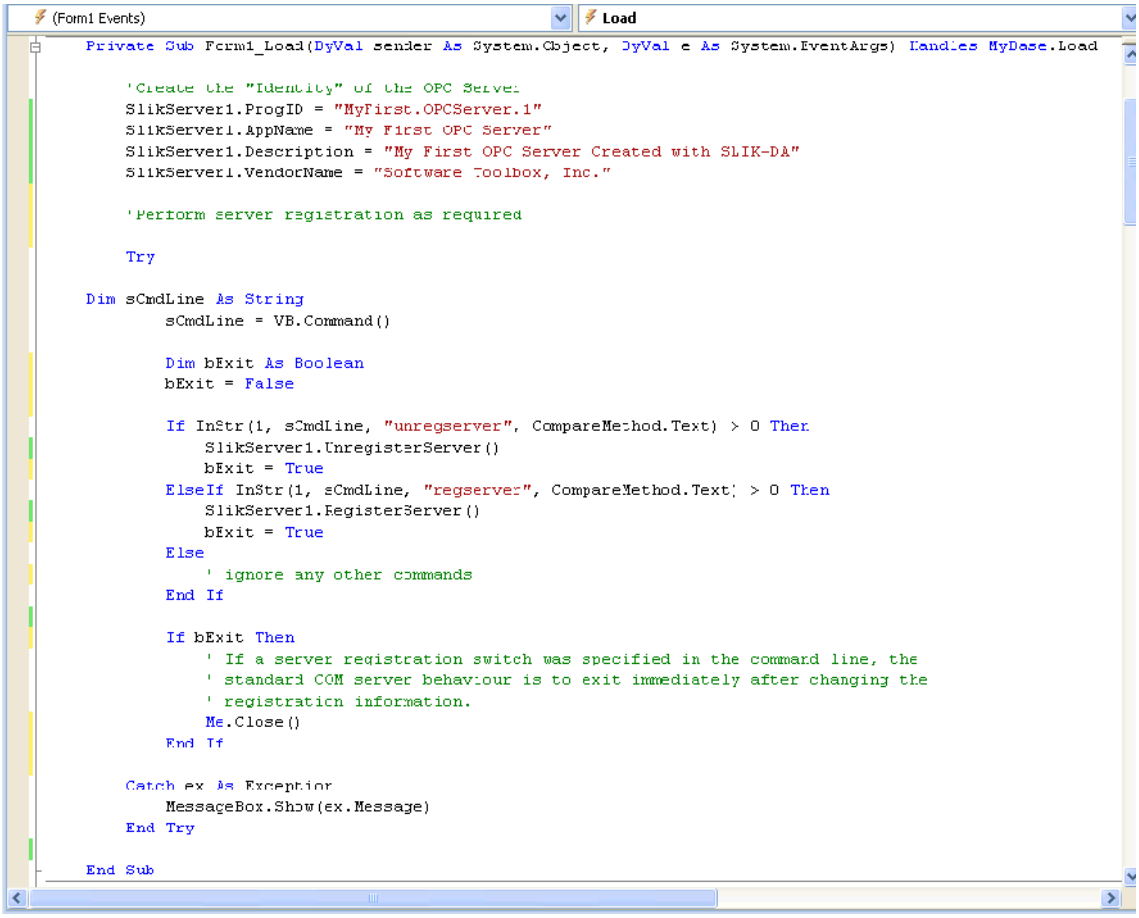


Figura 28 – Exemplo de linha de comando

Fonte: SLIKDA Tutorial

Neste momento é realizada a compilação do programa para gerar um arquivo executável. Após essa etapa deve-se atualizar o registro do servidor, para isso deve-se ir para o menu iniciar do Windows e clique em executar,

navegue e selecione o arquivo executável que acabou de ser criado, como exemplificado na figura 29.

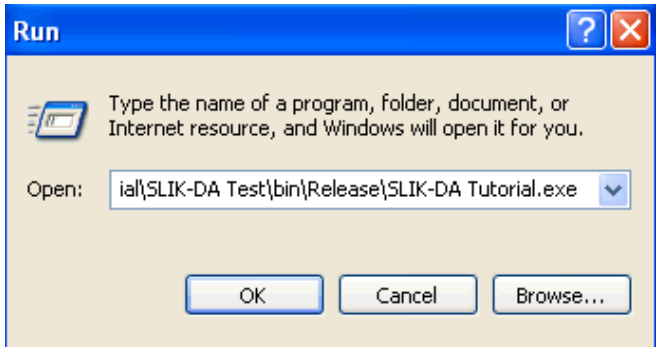


Figura 29 – Busca do arquivo executavel criado

Fonte: SLIKDA Tutorial

Para registrar o servidor é necessário adicionar uma linha de comando para registra-lo, neste caso utiliza-se “regsserver”(Figura 30), se for necessário retirar o registro pode-se utilizar “unregserver”. Para finalizar este processo clique em OK.

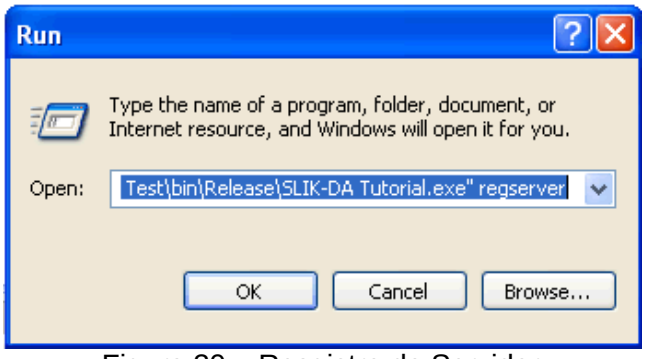


Figura 30 – Resgistro do Servidor

Fonte: SLIKDA Tutorial

Para verificar se o Servidor OPC está instalado no servidor pode-se testar utilizando um OPC Cliente e buscar pelos servidores disponíveis no computador.

Após instalação do Servidor é necessário configurar as tags que o servidor OPC vai expor ao público, para isso é criada uma linha de comando

para Criar as tags. Um exemplo da linha de comando que pode ser utilizada para criar as tags é apresentado na figura 31.

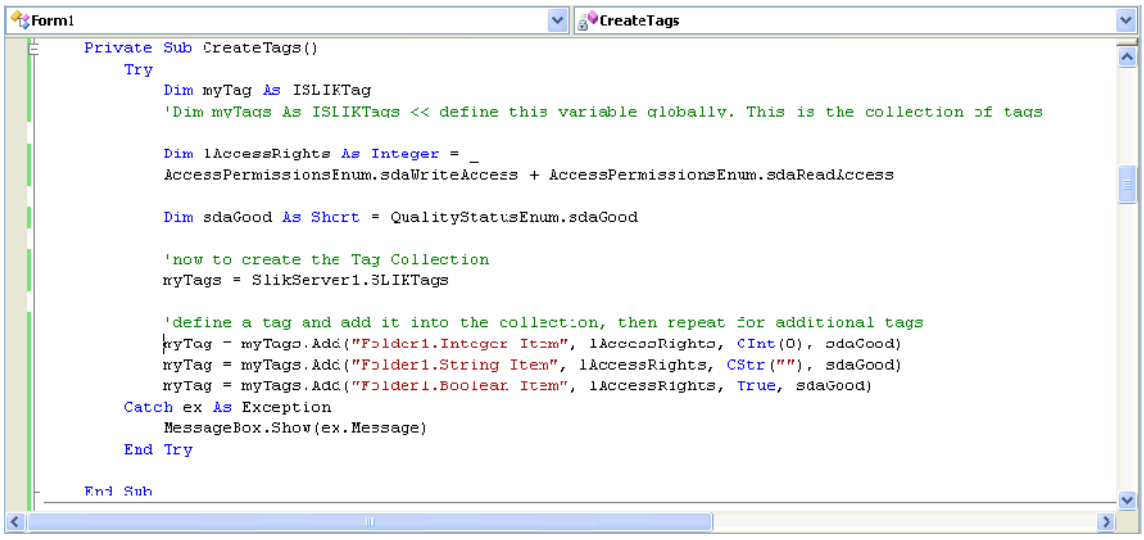


Figura 31 – Criação de Tags

Fonte: SLIKDA Tutorial

Da mesma forma utilizada anteriormente, para verificar se as tags foram realmente criadas pode-se testar utilizando um OPC Cliente e buscar pelas tags criadas.

Neste ponto não podemos escrever ou ler as tags, elas estão simplesmente sendo expostas, para ser possível a realização da leitura das tags é necessário adicionar uma linha de comando adicionando um script de tratamento(Figura 32). Este evento é disparado quando um cliente OPC solicita ler algum dispositivo (IOPSyncIO :: Read ()) com qualquer número de Tags. Portanto, este evento abriga uma coleção de tags que serão passados de volta diretamente para o cliente. Da mesma forma é necessário criar um script para realizar a escrita nas tags.

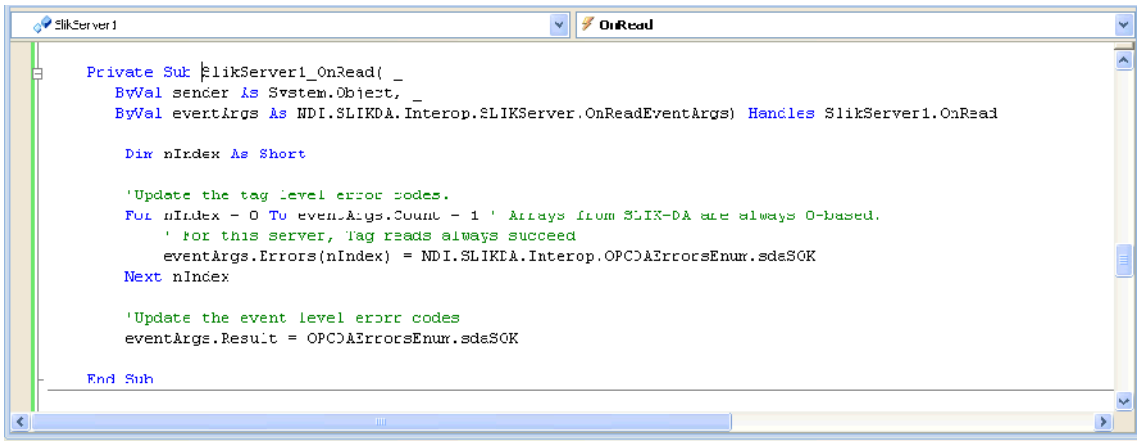


Figura 32 – Criando um procedimento para leitura das tags

Fonte: SLIKDA Tutorial

Apesar de fornecidos por empresas que se esmeram na confecção de seus sistemas, os *toolkits* não oferecem garantia de conformidade com o padrão OPC. Dessa forma, é necessário que se usem ferramentas de teste de conformidade independente do uso ou não de *toolkits* na geração de um servidor OPC.

7.3. Testes dos servidores

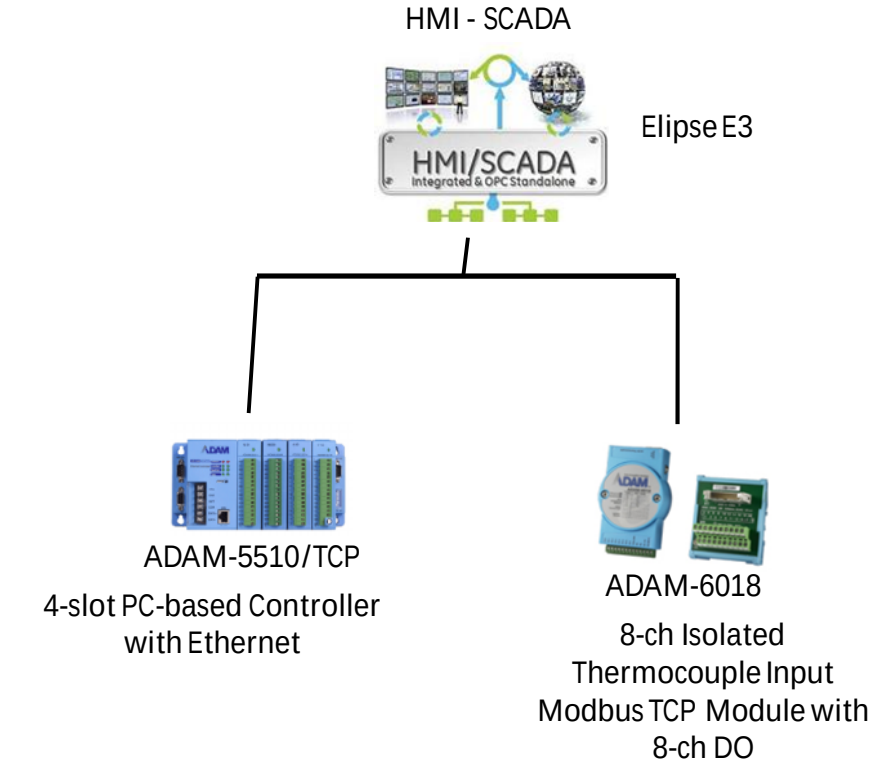
Para garantir que os servidores realizem as funções propostas no padrão OPC, é necessário verificar sua correta adequação às especificações. É necessário realizar testes que validem as interfaces de comunicação através de ferramentas específicas, desenvolvidas pelo *OPC Foundation*. Também é necessário garantir a qualidade do desenvolvimento realizado com o fim de recuperar dados e enviar comandos para os equipamentos de campo. Nesse caso, não é suficiente o uso de ferramentas automatizadas, mas sim o uso de sistemas supervisórios comerciais junto à solução, de forma a verificar se a os resultados alcançados podem ser utilizados em ambiente industrial. Neste trabalho foi utilizado o Elipse E3.

Para aquisição de dados em campo e interfaceamento com o sistema supervisório foram utilizados módulos de aquisição de dados da *Advantech - ADAM (Advantech Acquisition Modules)*. A escolha desses módulos foi baseado nas seguintes características (FRANCO;LENARTH,2010):

- Compatíveis com o Padrão OPC.
- Comunicam-se com padrões abertos RS 485, Ethernet, Modbus ou sem fio.
- Compatíveis com softwares abertos de desenvolvimento (Visual Basic, Visual C,..).
- Compatíveis com softwares proprietários (Intouch, Elipse, FIX, ...)

7.4. Ambiente de testes

Os testes foram realizados utilizando-se um computador ligado em rede, *MODBUS TCP*, com os módulos de Automação da empresa *Advantech – ADAM.* Foram utilizados os módulos *ADAM 5510/TCP* (Controlador Programável), *ADAM 6018* (Módulo de Entrada Termopar com 8 saídas digitais) e o *ADAM 5055S* (Módulo de 16 I/O Digitais), além de um Termopar Tipo J (0 a 200ºC). O banco de dados, o supervisório Elipse E3 e as ferramentas de checagem de conformidade foram instalados no computador (Figura 33 e 34). O Elipse E3 foi utilizado para supervisionar os resultados provenientes do servidor OPC DA. e OPC UA.



.

Figura 33 - Ambiente de Testes

Fonte: Elaborado pelo Autor

Figura 34 - Ambiente de Testes - Real

Fonte: Elaborado pelo Autor

7.5. Testes de conformidade

Os testes de conformidade (*Compliance Test*) fornecem a segurança sobre a adesão do servidor desenvolvido ao padrão OPC. Isso garante que o sistema se comunicará adequadamente com sistemas supervisórios que ofereçam clientes no padrão OPC. O objetivo é averiguar se as interfaces criadas são adequadas para o fim proposto.

Para realização desses testes é necessária uma série de ferramentas fornecidas pela *OPC Foundation*. Essas ferramentas trabalham sobre o servidor criado, permitindo averiguar quais interfaces não responderam de acordo com o padrão definido de respostas.

Para configurar o CTT é necessário definir os parâmetros de teste que deverão ser utilizados, de forma geral o CTT já possui alguns parâmetros “default” que podem ser alterados conforme necessidade do desenvolvedor. Neste trabalho foram utilizados todos os parâmetros “default”.

Na aba “General” além de buscar o servidor, defini-se também o número de testes que devem ser realizados, o número de testes em paralelo em cada operação e a frequência em que os casos de teste serão executados. Neste trabalho foram utilizadas as configurações apresentadas na Figura 35.

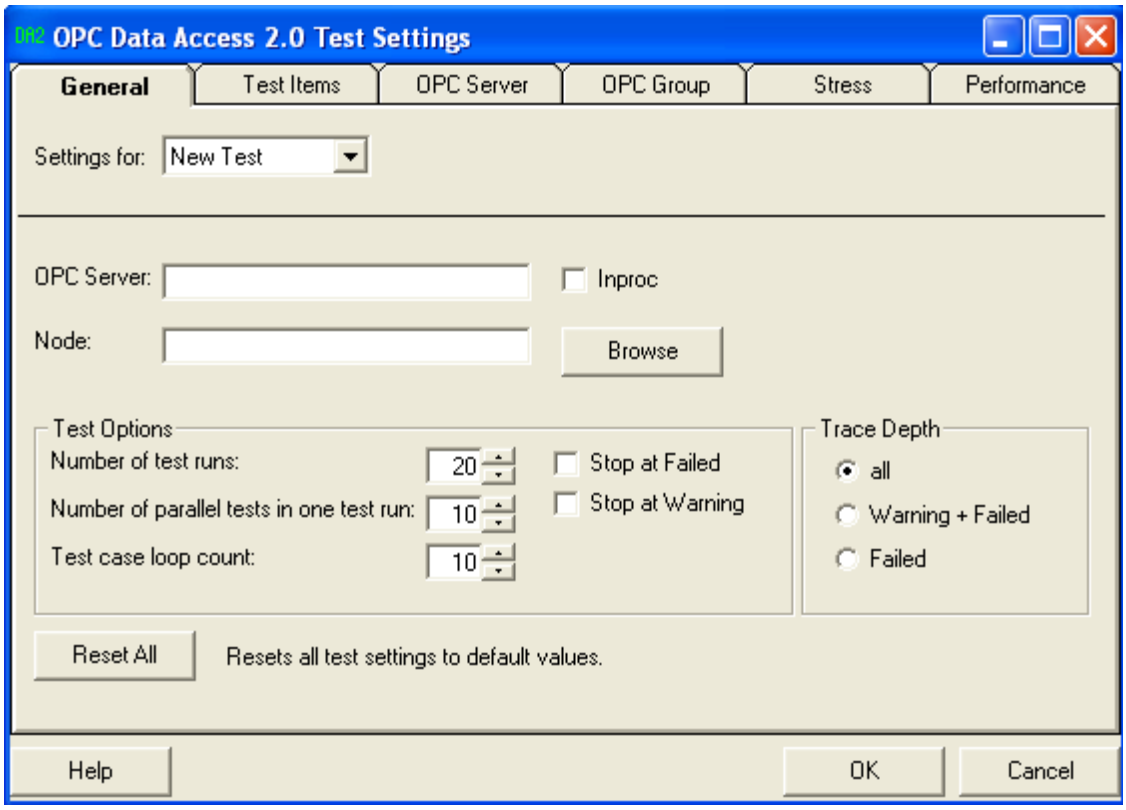


Figura 35 – Configuração Aba “General” CTT

Fonte: Elaborado pelo Autor

Na aba “OPC Server” defini-se os itens que devem ser usados no teste da interface do “IOPCItemProperties”, o número mínimo de “LocaleIDs” suportados, se o servidor não suporta o número de “LocaleIDs” são apresentadas falhas durante a execução, além disso é definido também o número de grupos criados para o teste. Neste trabalho foram utilizadas as configurações apresentadas na Figura 36.

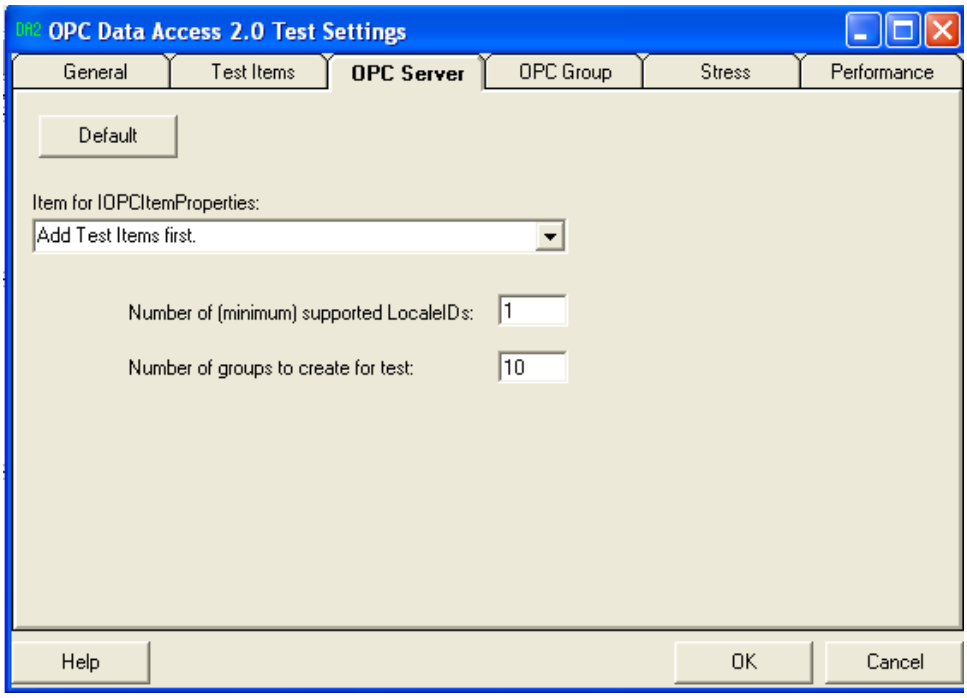


Figura 36 – Configuração Aba “OPC Server” CTT

Fonte: Elaborado pelo Autor

Na aba “OPC Group” defini-se os ItemID inválidos para o testes de erro, e o número de segundos de espera para o “Callback”. Neste trabalho foram utilizadas as configurações apresentadas na Figura 37.

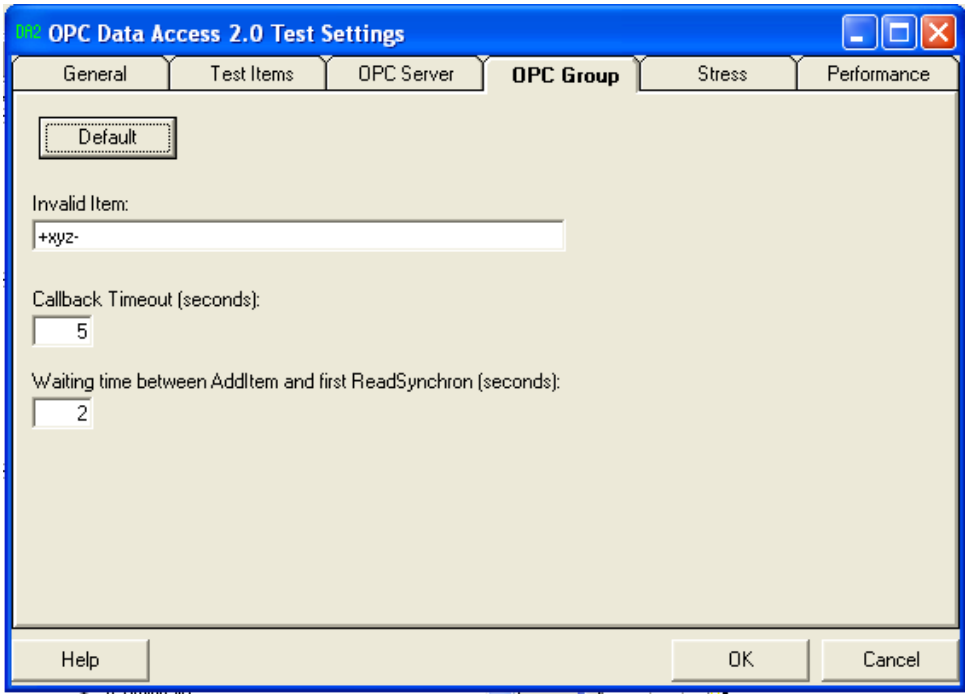


Figura 37 – Configuração Aba “OPC Group” CTT

Fonte: Elaborado pelo Autor

Na aba “Stress” são definidos o número de objetos que são criados para o testes de Stress, o número de grupos que são adicionados em cada objeto e o número de itens que são adicionados em cada objeto. Neste trabalho foram utilizadas as configurações apresentadas na Figura 38.

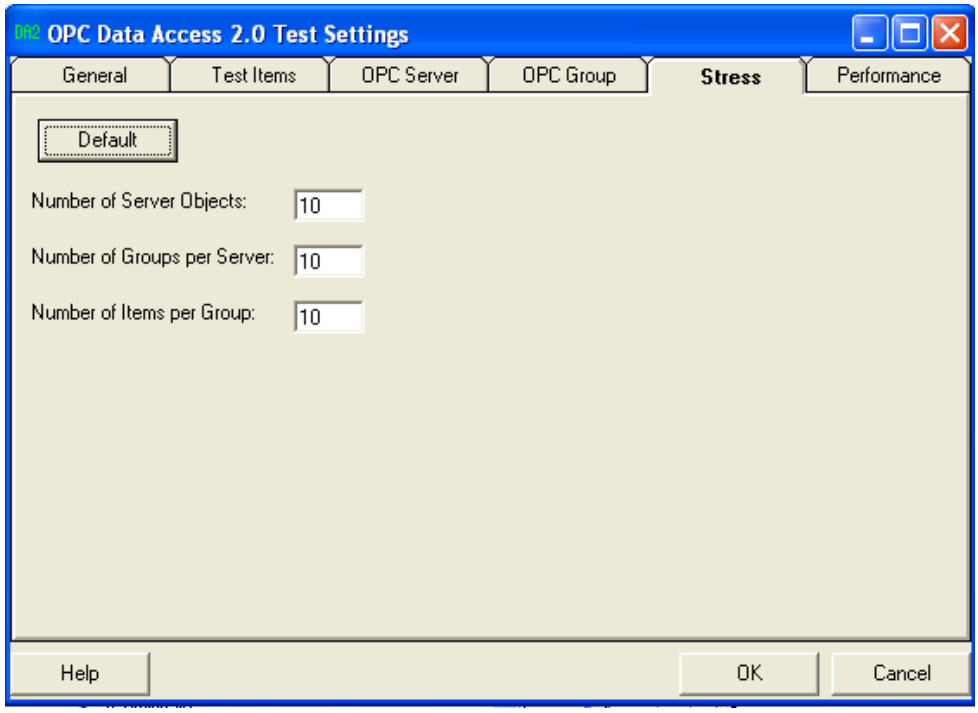


Figura 38 – Configuração Aba “Stresss” CTT

Fonte: Elaborado pelo Autor

Na aba “Performance” são definidos o número de itens que são usadas para o método de chamadas, a base de tempo em que os métodos especiais devem ser chamados e o número de chamadas que cada método deve ser chamado. Neste trabalho foram utilizadas as configurações apresentadas na Figura 39.

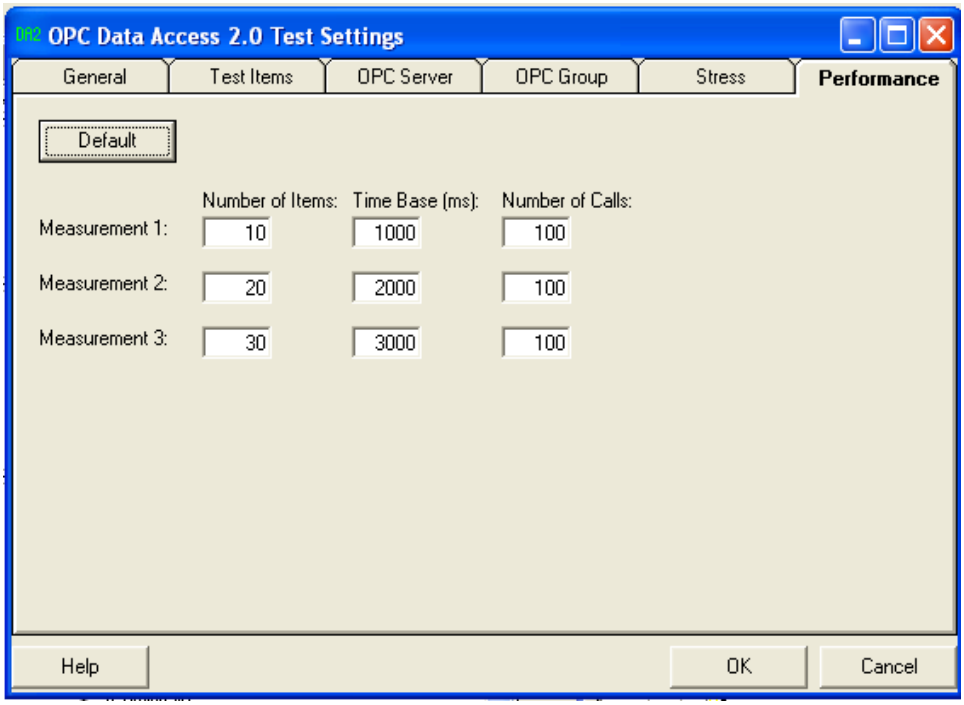


Figura 39 – Configuração Aba “Performance” CTT

Fonte: Elaborado pelo Autor

Durante os testes realizados com o Servidor OPC DA (Figura 40), verificou-se que três funcionalidades da interface gerada apresentaram falhas durante os testes. As funcionalidades com problemas foram o *IOPCBrowseServerAddressSpace* e o *IOPCItemProperties* do *OPCServerObject* e o *IOPCGroupStateMgt*, do *OPCGroupObject*.

O *IOPCBrowseServerAddressSpace* é uma interface opcional que fornece uma maneira para os clientes procurarem os itens de dados disponíveis no servidor, dando ao usuário uma lista de definições válidas para um Id Item. Ela fornece um espaços de endereços (Lista ou com hierarquia) . A apresentação hierárquica do espaço de endereço do servidor se comporta como um sistema de arquivos, onde os diretórios são os ramos ou caminhos, e os arquivos representam os itens. Um exemplo, seria um servidor apresentar um sistema de controle, mostrando todas as redes de controle, posteriormente todos os dispositivos na rede selecionada e todas as classes de dados dentro

de um dispositivo, em seguida, todos os itens de dados da classe. A falha estava na lógica de movimentação desses espaços de endereços hierárquicos, o que provocava travamentos no servidor.

O *IOPCItemProperties* é uma interface utilizada por clientes para verificar as propriedades disponíveis (atributos ou parâmetros) associada a um ItemID e ler os valores atuais dessas propriedades, Esta interface é baseada no pressuposto que um ItemID muitas vezes está associado com outros ITEMIDs , que representam os valores relacionados, unidades de Engenharia (controladores PID, temporizadores,...) ou descrição de alarme (Alarme deLimite Baixo ou Alto, ...), a falha estava na adição de *Items*, o que causava falha do servidor.

O *IOPCItemMgt* permite que um cliente adicione, remova e controle o comportamento de itens em um grupo, a falha na lógica ocorria quando um cliente tentava adicionar um novo item ao grupo.

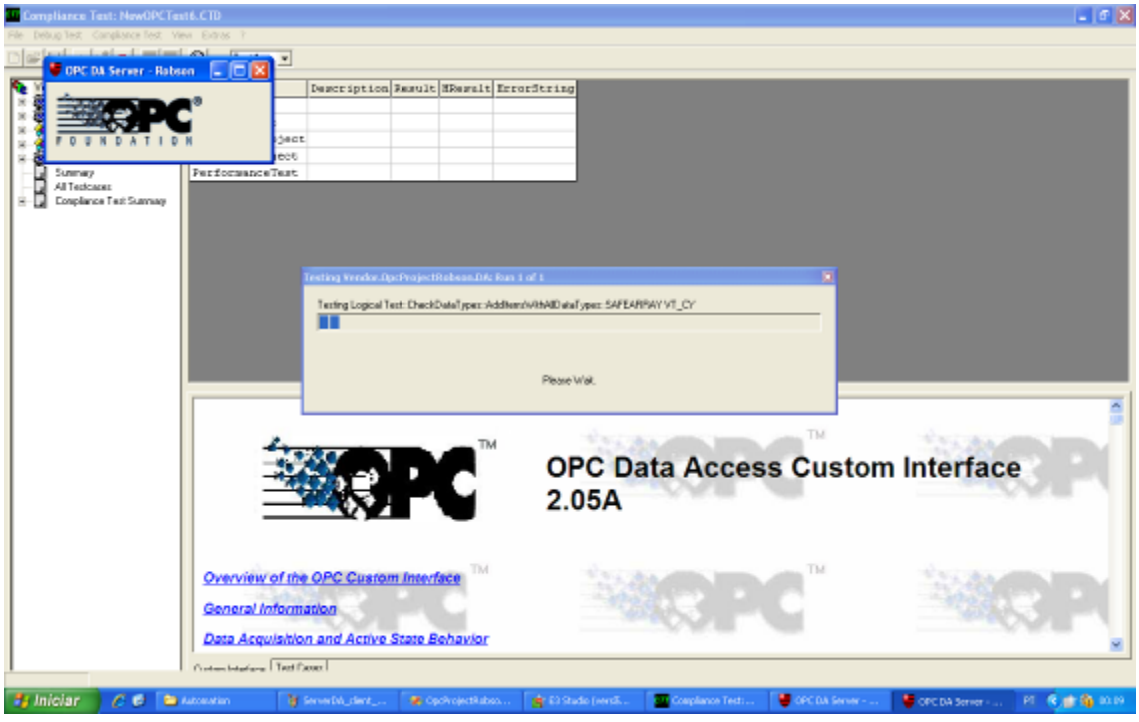


Figura 40 - CTT Servidor OPC DA Desenvolvido

Fonte: Elaborado pelo Autor

Para o CTT OPC UA é um "container" de livrarias de scripts de teste. Cada scripts de teste pode ser individualmente executado para uma funcionalidade específica do OPC UA. A importância dos testes é provar que o produto (Cliente ou Servidor) atende as especificações do OPC UA. Principalmente as seguintes especificações:

- OPC UA specifications (Parte 4)
    - Específica os Serviços fornecidos pelos servidores do OPCUA.
- OPC UA Profile (Partes 7 e 8)
    - Parte 7 – Introduz o conceito de perfil e define perfis disponíveis que são grupos de serviço ou funcionalidade.
    - Parte 8 – Específica o uso do OPC UA para “Data Access”.

Durante os testes realizados com o Servidor OPC UA (Figura 41), foram verificadas várias falhas inicialmente, desde endereçamento da url até travamento no servidor. Porém, a grande vantagem do CTT UA é a apresentação clara das falhas no próprio código realizado, auxiliando a correção.

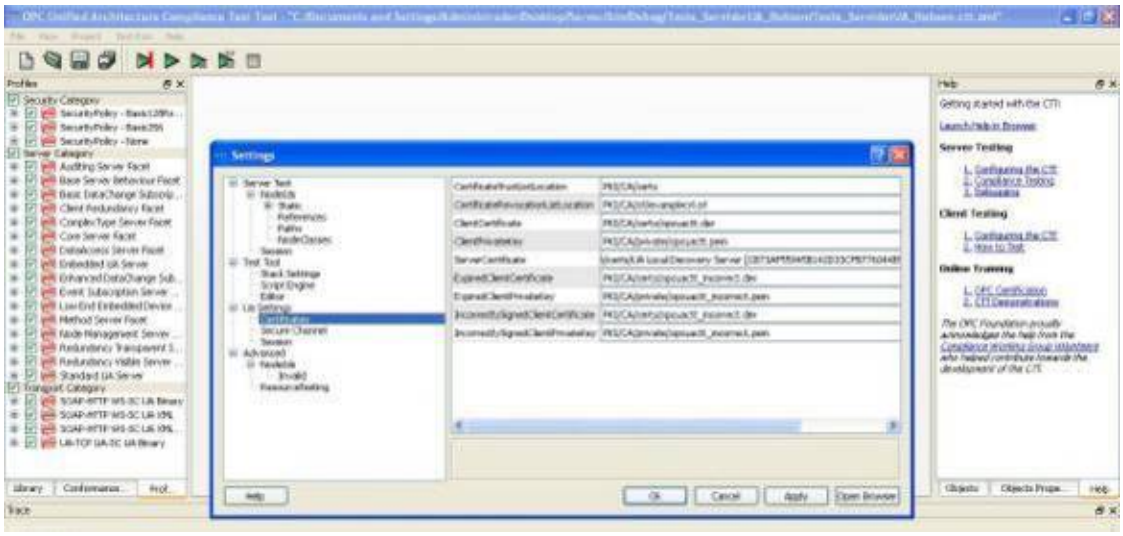


Figura 41 - CTT Servidor OPC UA Desenvolvido

Fonte: Elaborado pelo Autor

7.6. Testes com clientes

Os testes com os servidores foram feitos inicialmente a partir da simulação de comunicações cliente/servidor, realizada por aplicações clientes fornecida para testes pela Softing e OPC Foundation (Figura 42).

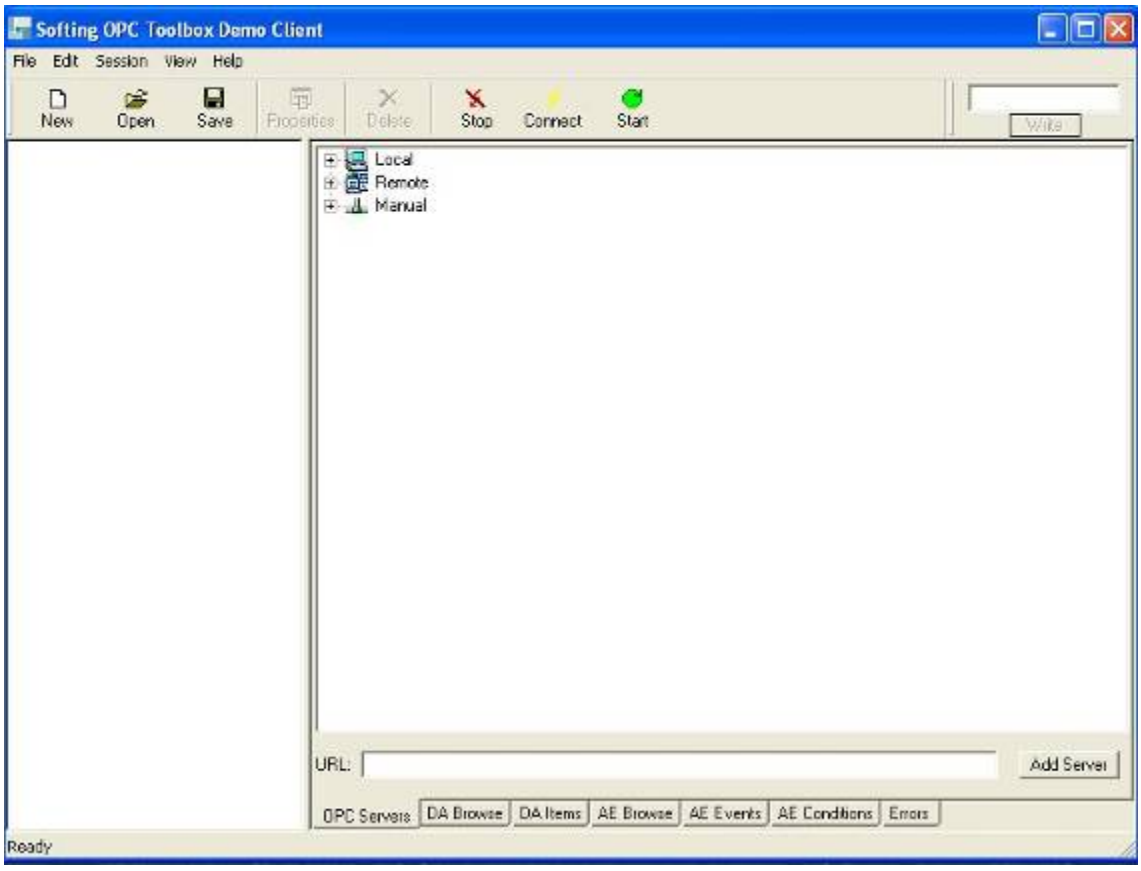


Figura 42 - Exemplo Cliente OPC – Softing

Fonte: Elaborado pelo Autor

Os clientes foram instalados no PC para verificar a funcionalidade de identificação do servidor OPC DA e OPC UA (Figura 43). Não foram encontrados problemas relacionados a essas funcionalidades, pois já haviam sido realizados os testes de conformidade.

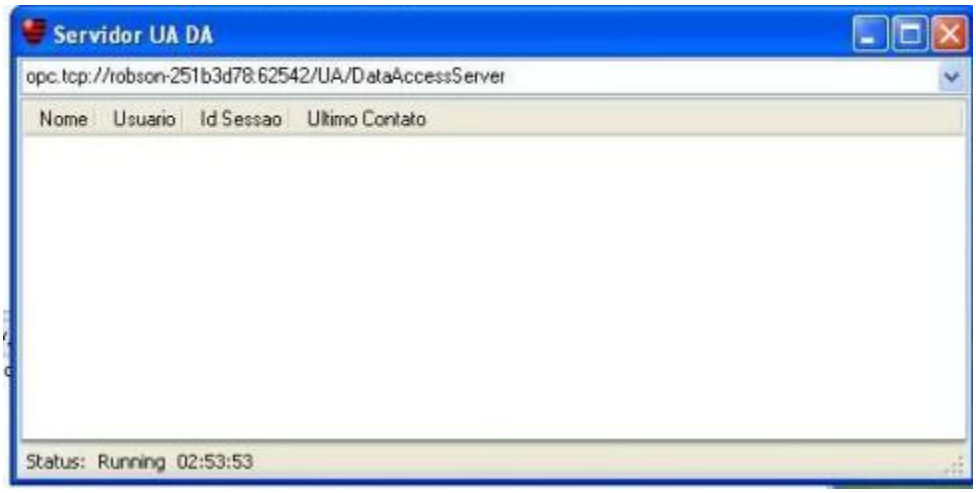


Figura 43 - Servidor OPC UA e OPC DA

Fonte: Elaborado pelo Autor

Posteriormente foi criada uma interface de supervisão e controle no Elipse E3 (Figura 44), realizada a configuração do Elipse E3 para comunicação com o Servidor OPC (Figura 45) desenvolvido, e posteriormente realizado os testes da aplicação em tempo real.

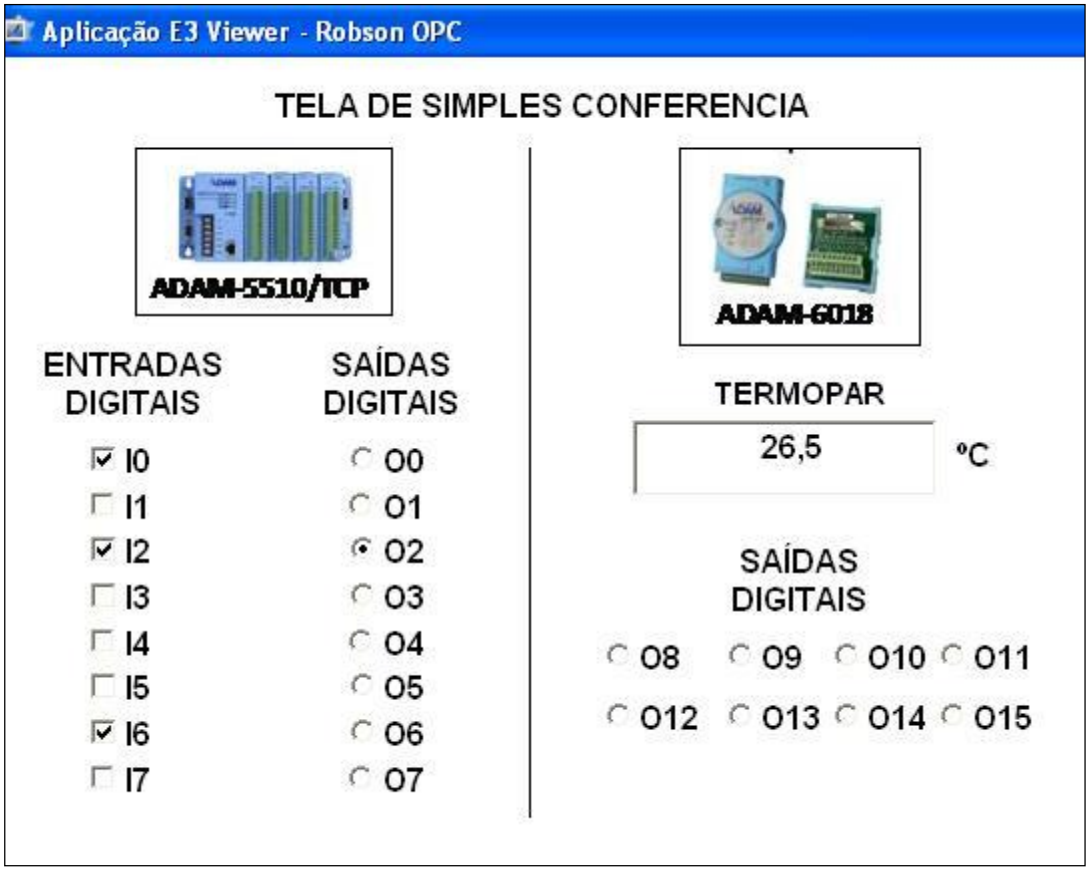


Figura 44 - Interface desenvolvida

Fonte: Elaborado pelo Autor

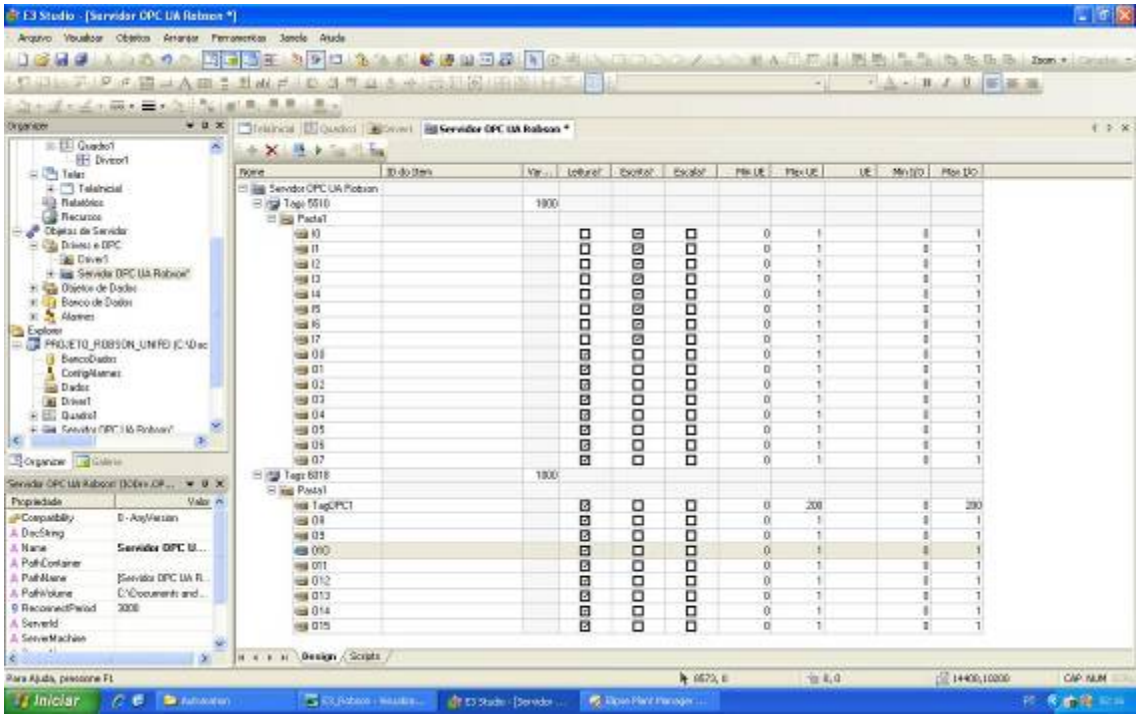


Figura 45 - Elipse E3 - Configuração OPC

Fonte: Elaborado pelo Autor

8. DESENVOLVIMENTO WRAPPER OPC

8.1. Ambiente de desenvolvimento

Para desenvolver os sistemas foi necessário o uso de ferramentas de programação de alto nível como descrito no capitulo anterior. Da mesma forma foi utilizado o *MS Visual Studio*, essencial para o desenvolvimento de cada módulo utilizado nos *Wrappers*, Para o desenvolvimento do *Wrapper OPC* foi utilizado um toolkit padrão, fornecido pela *OPC Foundation*.

8.2. Ambiente de testes

Os testes foram realizados acrescentando-se o software Elipse Plant Manager – EPM ao conjunto apresentado no capitulo 8.4 (Figura 46).

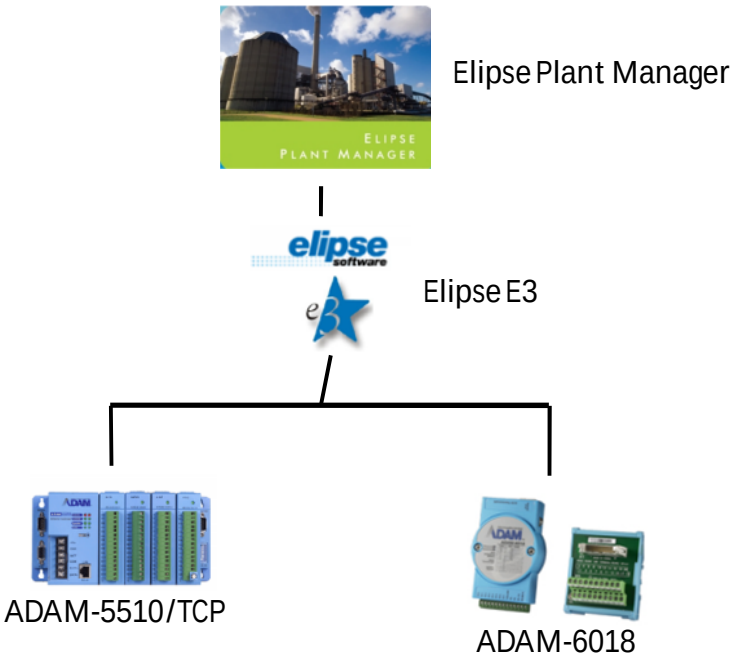


Figura 46 - Ambiente de Testes com EPM

Fonte: Elaborado pelo Autor

O EPM foi utilizado para aquisitar os resultados provenientes do servidor

OPC DA (Desenvolvido), proveniente da ELIPSE E3, e ser entendido por um OPC UA Cliente que é representado pela EPM. Para isso foi desenvolvido o OPC *Wrapper*, permitindo clientes UA acessar os dados de qualquer fornecedor em uma interface padrão de OPC DA Clássico.

8.3. Testes Wrapper OPC

Com o desenvolvimento do *Wrapper* OPC foi criada uma interface de supervisão no Elipse EPM (Figura 49), realizada a configuração do Elipse EPM para comunicação com o Servidor OPC DA *Wrapper* (Figura 47) desenvolvido, e posteriormente realizado os testes da aplicação em tempo real.

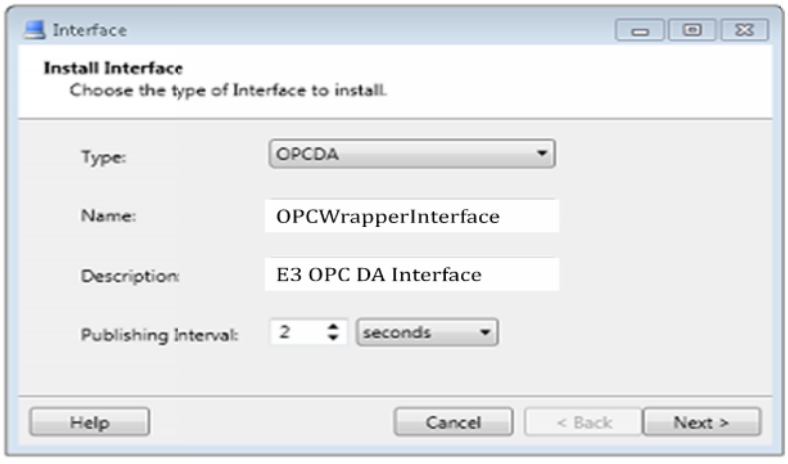


Figura 47 - EPM – Configuração OPC

Fonte: Elaborado pelo Autor

Na caixa de seleção Type, deve-se selecionar o tipo de Interface de Comunicação OPCDA, o campo Publishing Interval corresponde ao intervalo de tempo antes do qual não serão recebidos dados do Servidor OPC-DA. Finalizando essa configuração deve-se clicar em “Next” (Tutorial Elipse Plant Manager, 2011). Na Janela “Interface”, mas especificamente na caixa de

listagem aparecerão todos os Servidores disponíveis no computador local.

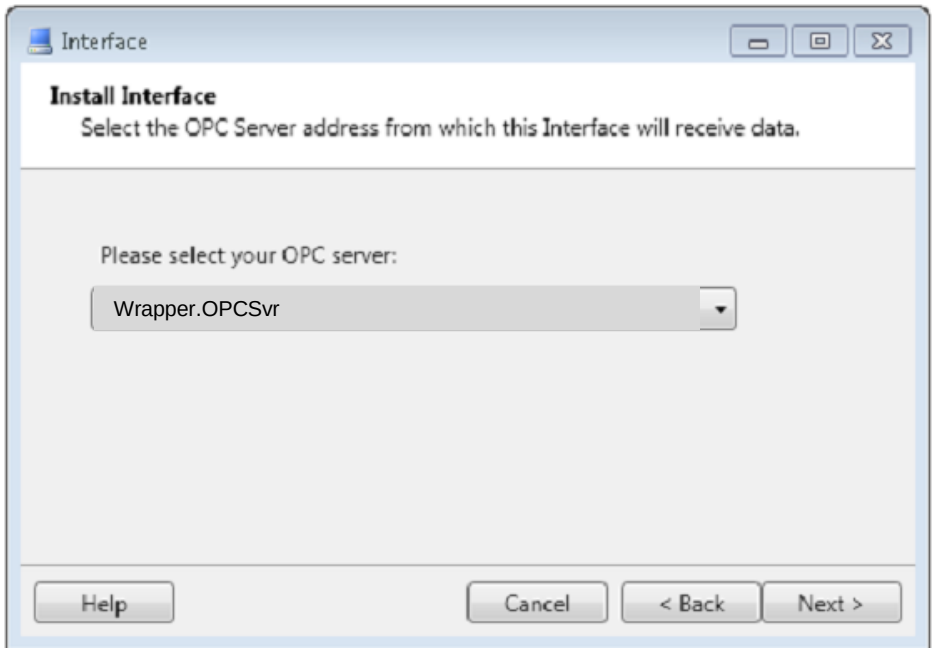


Figura 48 - EPM – Configuração Interface OPC

Fonte: Elaborado pelo Autor

Uma vez instalada e configurada a Interface de Comunicação com um Servidor, é preciso definir as Tags que estarão vinculados a ela. Após criada a Interface, é necessário também selecionar o servidor ou servidores E3 ou Elipse Power que serão monitorados.

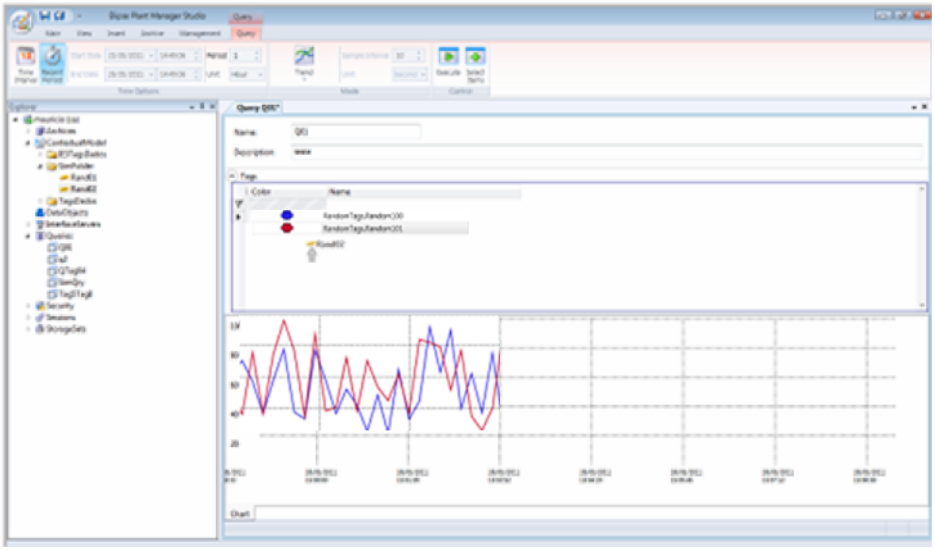


Figura 49 - EPM – Desenvolvido

Fonte: Elaborado pelo Autor

Algumas falhas ocorreram durante os testes, uma delas foi o travamento do servidor *OPC Wrapper*, devido principalmente a uma falha na lógica do OPC *Wrapper*, na comunicação entre o *Monitored Items* e *OPC Items*.

Outro ponto de falha foi o atraso na aquisição de dados e monitoramento de dados devido às falhas de configuração do EPM. Porém, a grande vantagem do CTT UA é a apresentação clara das falhas no próprio código realizado, auxiliando a correção das falhas presentes no Wrapper.

9. **CONCLUSÕES**

Inicialmente foi realizado um estudo sobre a tecnologia do OPC Clássico e UA, além de esclarecer conceitos relevantes para o desenvolvimento de servidores, bem como os testes necessários.

Foram desenvolvidos Servidores OPC DA e UA com uma comunicação confiável para aquisição de informações, disponibilizando-as de forma padronizada, conforme especificações da OPC foundation, garantida pelos

testes realizados utilizando o CTT. Para uma aplicação comercial os servidores desenvolvidos deverão passar por uma homologação da OPC Foundation, em seus laboratórios.

Durante o desenvolvimento do servidor OPC DA fica nítido uma facilidade, devido à grande quantidade de *toolkits* disponíveis para o desenvolvimento e a maturidade da tecnologia utilizada. Os toolkits possuem a característica de um toolbox, possuindo um conjunto de funções/blocos que facilitam a o desenvolvimento das linhas de comando.

A ampliação do OPC UA entre os servidores de dados, as vantagens associadas à interoperabilidade UA junto com a criação de novas ferramentas de apoio ao desenvolvimento, e uma melhor definição das questões de segurança, item que não foi abordado nesse trabalho, mas que é de grande relevância fará desse padrão a melhor solução para uso na indústria.

Foi desenvolvido um Servidor Wrapper capaz de intermediar a comunicação entre as duas tecnologias OPC, realizando o acesso a um Servidor COM através de um Cliente UA, Porém a interface apresentou inicialmente um funcionamento instável, devido a falhas na configuração e implantação, principalmente no início dos testes. Porém, utilizando o CTT UA foi possível corrigir o código, auxiliando a correção das falhas presentes no Wrapper.

Com este trabalho foi realizado uma integração em todos os níveis da pirâmide da Automação, deste do nível de campo até o nível de controle e supervisão. Visualizando na prática todos os conceitos teóricos envolvidos, bem como dificuldades inerentes a atividades práticas, onde novos caminhos e

desafios são trilhados a cada novo passo do projeto, e que por muitas vezes ultrapassa o meio acadêmico, principalmente na aquisição de hardwares (Instrumentação, Contralador e Módulos de Aquisição de Dados),e pedidos de doação de softwares(OPC Foundation).

Este trabalho permitiu unir todo o conhecimento técnico adquirido durante o curso à experiência prática, visualizando claramente as dificuldades enfrentadas durante o cotidiano no desenvolvimento de projetos, desde o desafio técnico até as questões financeiras e burocráticas.

Para trabalhos futuros aprimoramentos serão realizados na interface OPC Wrapper e tem-se como sugestão o desenvolvimento de um Servidor UA Híbrido, com o objetivo de incorporar o Wrapper e Proxy no próprio Servidor UA, retirando um componente adicional entre um cliente OPC (UA or DCOM) e um Servidor OPC (DCOM ou UA). Esta adição eliminaria um possível ponto de falha, que é a transição do cliente/servidor com o Wrapper/Proxy, aumentando a segurança, reduzindo as fontes de erro além de garantir a interoperabilidade da tecnologia OPC.

REFERÊNCIAS

ABDMOULEH, A.; SPADONI, M.; VERNADAT, François; "***Distributed client/server architecture for CIMOSA-based enterprise components***", Computers in Industry, 2004.

ALBUQUERGUE, Pedro Urbano Braga de; **Redes industriais: Aplicações em Sistemas Digitais de Controle Distribuido**, Edições Livro Técnico, Fortaleza, 2007.

BAILEY, David; WRIGHT, Edwin; ***Practical SCADA for Industry***, Newnes an imprint of Elsevier, Great Britain, 2003.

BOYER, S. A., "***SCADA: Supervisory Control and Data Acquisition***", ISA Books, 2004.

BURKE, J. Thomas; IWANITZ, Frank; LANGE, Jurgen; ***OPC From Data Access to Unified Architecture***, 4ª Rev. Ed., VDE VERLAG GMBH, Berlin – Offenbach, 2010.

CLARKE, Gordon; REYNDERS, Deon; WRIGHT, Edwin; ***Practical Modern SCADA Protocols***: *DNP3, 60870.5 and Related Systems*, Newnes An imprint of Elsevier, Great Britain, 2004.

DIYTRADE, **Global B2B Trading Platform**. Disponível em: <http://www.diytrade.com/china/4/products/7471320/Allen_Bradley_Panel_View_1000.html>. Acesso em: 11 dez. 2011.

ELIPSE SOFTWARE, **Elipse Plant Manager – Descrição**,, http://www.elipse.com.br/produto_texto.aspx?id=22&opcao=101&idioma=1 , acessado em novembro de 2011.

FRANCO**,** Lúcia R. H.; LENARTH, Luiz; **Curso de Redes Industriais** – FUPAI – Fundação de Pesquisa e Assessoramento à Indústria de Itajubá. Itajubá – MG - Brasil, Setembro de 2010.

HMS ANYBUS, **Modbus-IDA Ethernet TCP/IP Protocol**. Disponível em: < http://www.hms.se/technologies/modbustcp.shtml>. Acesso em: 11 dez. 2011.

IWANITZ, F.; Lange, J.; ***OPC Fundamentals, Implementation, and Application***, 2ª Ed., Hüthig Verlag Heidelberg, 2002

KICHALOWSKY, Marco Andrei, **E3 - Uma visão geral**, Elipse Software, http://kb.elipse.com.br/pt-br/questions/40/E3+-+Uma+vis%C3%A3o+geral, 17th of December, 2010.

LIMA,Samir**; Arquitetura de Sistemas de Supervisão,** Elipse Software Disponível em: <www2.fc.unesp.br/wes/materias/IV/**elipse**.ppt>*,* Acesso em: 11 dez. 2011.

MAHNKE, Wolfgang; LEITNER, Stefan-Helmut; DAMM, Matthias; ***OPC Unified Architecture***, Ed. Springer - Verlag, Berlin – Heidelberg, 2009.

MORAES, Cícero Couto De; CASTRUCCI, Plínio De Lauro; **Engenharia De Automação Industrial**, LTC - Livros Técnicos e Científicos Editora, 2ª Ed, Rio de Janeiro, 2007.

NASCIMENTO FILHO, Osmar A., **Desenvolvimento de Servidores OPC DA e OPC XML DA para Sistemas de Aquisição de Dados via Telefonia Celular**, Vitória, 2005.

OPC COMMON DEFINITIONS AND INTERFACES, Version 1.0,OPC Foundation, October 1998.

OPC DATA ACCESS CUSTOM INTERFACE STANDARD ,Version 2.05A, OPC Foundation, June 2002.

OPC DATA ACCESS CUSTOM INTERFACE STANDARD, Version 3.00, OPC Foundation, March 2003.

OPC TEST LAB SPECIFICATION: PART 1 – CONCEPTS, OPC Foundation, January 2008.

OPC TEST LAB SPECIFICATION: PART 2 – ABSTRACT TEST SUITE, OPC Foundation, January 2008.

OPC TEST LAB SPECIFICATION: PART 3 – TEST LAB REALIZATION, OPC Foundation, January 2008.

OPC TEST LAB SPECIFICATION: PART 4 – CERTIFICATION PROCESS, OPC Foundation, January 2008.

OPC UA PART 1 - OVERVIEW AND CONCEPTS 1.01 SPECIFICATION, OPC Foundation, February 2009.

OPC UA PART 2 – SECURITY MODEL 1.01 SPECIFICATION, OPC Foundation, February 2009.

OPC UA PART 3 – ADDRESS SPACE MODEL 1.01 SPECIFICATION, OPC Foundation, February 2009.

OPC UA PART 4 – SERVICES 1.01 SPECIFICATION, OPC Foundation, February 2009.

OPC UA PART 8 – DATA ACCESS 1.01 SPECIFICATION, OPC Foundation, February 2009.

OPC FOUNDATION; ***Compliance Test Tool – CTT***; *GUI* Versão: 2.15.0.1152, *CTT Data Control* Versão: 2.00.15.1152, OPC Data Access 2.05a *Compliance Test* Versão 2: *Test Control* Versão: 2.00.20.1156 e *Test Settings* Versão: 2.20.0.1156, 2011.

RIEDL, M.; THRON, M.; HADLICH, T., "***DriveServer - Significantly reduce in engineering expense***", IECON'01: The 27th Annual Conference of the IEEE Industrial Electronics Society, 2001.

SLIK-DA Tutorial, Creating a simple server in Visual Studio, Software Toolbox. Disponível:<http://support.softwaretoolbox.com/ci/fattach/get/34384/130512224 4/redirect/1/session/L2F2LzEvdGltZS8xMzM1MjMzMzcyL3NpZC9DZDdvY3BX aw==/filename/SLIK-DA%20VS2008.pdf>. Acesso em: 23 abril. 2012.

PLÁSIL, F., STAL, M., ***An architecture view of distributed objects and components in CORBA, Java RMI and COM/DCOM***, *Software Concepts & Tools*, n. 19, p. 14 – 28, 1998.

SILVA, Ana Paula Gonçalves da; SALVADOR, Marcelo; **O que são sistemas supervisórios**?,, 21st of December, 2010.

TANENBAUM, A. S., ***Computer Networks***, Prentice Hall, 4ª Ed., 2003.

TUTORIAL ELIPSE PLANT MANAGER, Elipse Software, Ver. 1.0 ,05 de Outubro de 2011.