



**Universidade de
Aveiro**
Ano 2012

Departamento de Engenharia Mecânica

**Luís Carlos
da Costa
Figueiredo**

**Serviços WEB – Supervisão e controlo de
sistemas de produção**



**Luís Carlos
da Costa
Figueiredo**

**Serviços WEB – Supervisão e controlo de
sistemas de produção**

Tese apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em (Mestrado Integrado em Engenharia Mecânica), realizada sob a orientação científica do Professor Doutor José Paulo Santos, Professor Auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro e do Professor Doutor Joaquim José Borges Gouveia, Professor Catedrático do Departamento de Economia, Gestão e Engenharia Industrial da Universidade de Aveiro.

Dedico este trabalho à minha Família.

O júri

Presidente

Prof. Doutor António Gil d'Orey de Andrade Campos
Professor auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro

Arguente

Prof. Doutor João José da Cunha e Silva Pinto Ferreira
Professor associado do Departamento de Engenharia e Gestão Industrial da Faculdade de Engenharia da Universidade do Porto

Orientador

Prof. Doutor José Paulo Oliveira Santos
Professor auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro

Coorientador

Prof. Doutor Joaquim José Borges Gouveia
Professor catedrático do Departamento de Economia, Gestão e Engenharia Industrial da Universidade de Aveiro

Agradecimentos

Agradeço ao professor José Paulo Santos pelo acompanhamento contínuo durante o trabalho e pela ajuda prestada.

Agradeço ao professor Borges Gouveia pela colaboração e o apoio prestado.

Agradeço ao professor André Quintã pela ajuda prestada.

Agradeço aos meus pais e irmão pela compreensão, por todo o apoio prestado e pelas condições que me permitiram chegar aqui.

Agradeço a todos os meus colegas e amigos que sempre me acompanharam.

Obrigado a todos.

Palavras-chave

Serviços Web, SCADA, SOA, integração de sistemas, controle, monitorização.

Resumo

A constante criação de novos produtos provoca mudanças constantes no processo produtivo levando muitas vezes a problemas de integração de recursos. Neste contexto a arquitetura SOA surge como uma interessante solução nomeadamente através da tecnologia dos serviços Web.

Por outro lado o controlo dos processos de produção torna-se cada vez mais fulcral para o processo produtivo sendo que uma das ferramentas mais amplamente usadas são os sistemas SCADA. Neste momento mesmo os sistemas SCADA encontram-se sob pressão para permitir o controlo e a monitorização dos processos de uma maneira mais fácil e eficaz permitindo ao mesmo tempo efetuar esse controlo de qualquer parte do mundo e que o mesmo esteja disponível 24 horas por dia.

Define-se então um novo tipo de solução SCADA que tem por ambição combinar as mais valias dos serviços Web e as ferramentas fornecidas por uma aplicação SCADA na medida em que beneficia a descentralização, independência e interoperabilidade do controlo dos processos a nível industrial.

A implementação experimental da solução tornou possível verificar a simplicidade e facilidade da comunicação entre o serviço Web e o recurso a monitorar bem como uma boa integração entre o serviço Web e a aplicação SCADA permitindo um controlo e monitorização eficazes.

Keywords

Web services, SCADA, SOA, systems integration, control, supervision.

Abstract

The constant creation of new products causes constant changes in the production process often leading to problems of resource integration. In this context, SOA emerges as an interesting solution in particular through the technology of Web services.

Moreover the control of production processes becomes increasingly critical to the production process of which one of the most widely used tools are SCADA systems. At this very moment SCADA systems are under pressure to allow control and monitoring of processes in a more easy and efficient way while allowing this control to be carried out from anywhere in the world and that it is available 24 hours a day.

Sets up a new kind of SCADA solution that has the ambition to combine the gains of Web services and tools provided by a SCADA application to the extent that benefits decentralization, independence and interoperability of process control at industrial level.

The experimental implementation of the solution made it possible to verify the simplicity and ease of communication between the Web service and resource monitoring as well as good integration between the Web service and SCADA application developed allowing effective control and monitoring.

Índice

Lista de Figuras	ii
Lista de Acrónimos	iii
1. Introdução.....	1
1.1 A necessidade do controlo industrial	1
1.2 Enquadramento	1
1.3 Objetivos	3
1.4 Organização da Dissertação	4
2. Serviços Web e sistemas SCADA.....	5
2.1 Serviços Web.....	5
2.2 Tecnologias de Suporte	8
2.2.1 SOAP	8
2.2.2 WSDL.....	11
2.2.3 UDDI.....	14
2.3 SCADA.....	17
2.3.1 Exemplo de um sistema SCADA.....	19
2.4 Alguns Trabalhos na Área	20
3. Solução.....	31
3.1 A crescente importância dos serviços Web.....	31
3.2 A importância de um sistema SCADA e as novas exigências do mercado	31
3.3 Solução encontrada	32
4. Implementação	35
4.1 Recurso.....	36
4.2 Serviço Web	37
4.2.1 Implementação	40
4.2.2 Descrição WSDL	40
4.2.3 Consumo	41
4.3 Aplicação SCADA.....	43
4.3.1 Integração do serviço Web.....	44
5. Considerações Finais	49
5.1 Trabalhos Futuros.....	50
Bibliografia	51

Lista de Figuras

Figura 1 – Protocolos nos quais assentam os serviços Web [Cerami, 2002].	7
Figura 2 – Tecnologias que compõem um serviço Web [Filho, 2007].	8
Figura 3 – SOAP na arquitetura dos serviços Web [WS-Tutorial, 2011].	8
Figura 4 – Estrutura de uma mensagem SOAP [Tidwell, 2001].	9
Figura 5 – Exemplo de um pedido SOAP com múltiplos parâmetros [Shklar, 2003].	10
Figura 6 – Resposta ao pedido SOAP anterior [Shklar, 2003].	11
Figura 7 - WSDL como um mecanismo para facilitar a comunicação entre serviços Web e outras aplicações [Daconta, 2003].	14
Figura 8 – As várias camadas de um serviço Web e o respetivo registo UDDI [Bellwood, 2002].	15
Figura 9 – Registo UDDI representado como uma lista telefónica [Daconta, 2003].	16
Figura 10 – Esquema de um sistema SCADA [Clarke, 2004].	18
Figura 11 – Exemplo de um sistema SCADA [SCADA - TechnologyUK].	20
Figura 12 – Solução proposta.	33
Figura 13 – Solução encontrada.	35
Figura 14 – Tapete de transporte presente no laboratório.	36
Figura 15 – Métodos do serviço Web.	39
Figura 16 – Código referente ao método "tapete_esquerda".	39
Figura 17 – Descrição WSDL do serviço "Tapete_Industrial".	41
Figura 18 – Mensagem SOAP para invocar o método "tapete_esquerda".	42
Figura 19 – Mensagem SOAP de resposta ao método "tapete_esquerda".	43
Figura 20 – Interface de controlo da aplicação SCADA.	44
Figura 21 – Inclusão da biblioteca de funções.	45
Figura 22 – Código do botão que movimenta o tapete para a esquerda.	46
Figura 23 – Rotina de comparação que possibilita a monitorização do recurso.	47

Lista de Acrónimos

API	Application Programming Interface
AVAC	Aquecimento Ventilação e Ar Condicionado
BAS	Building Automation Systems
BEEP	Blocks Extensible Exchange Protocol
CIM	Computer Integrated Manufacturing
CNC	Computer Numerical Control
CORBA	Common Object Request Broker Architecture
CPU	Central Processing Unit
DCS	Distributed Control System
DPWS	Devices Profile for Web Services
ERP	Enterprise Resource Planning
FTP	File Transfer Protocol
HMI	Human-Machine Interface
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IB	Intelligent building
IIS	Internet Information Services
ITEA	International Test and Evaluation Association
JNI	Java Native Interface
MES	Manufacturing Execution System
NAICS	North American Industry Classification System
ODBC	Open Database Connectivity
OLE	Object Linking and Embedding
OPC	OLE for Process Control
PLC	Programmable Logic Controller
RCP	Remote Procedure Call
RTU	Remote Terminal Unit
SCADA	Supervisory Control and Data Acquisition
SIC	Standard Industrial Classification
SMTP	Simple Mail Transfer Protocol
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SoS	System of Systems
SQL	Structured Query Language

UDDI	Universal Description Discovery and Integration
UNSPSC	United Nations Standard Products and Services Code
UPnP	Universal Plug and Play
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VBA	Visual Basic for Applications
W3C	World Wide Web Consortium
WML	Wireless Markup Language
WSDL	Web Service Description Language
XML	Extensive Markup Language
XSD	XML Schema
XSLT	Extensible Stylesheet Language Transformations

1. Introdução

1.1 A necessidade do controlo industrial

É imperativo para qualquer sistema industrial que haja um controlo do seu ambiente fabril, quer seja para identificar o número de peças produzidas num determinado período de tempo num determinado recurso, quer seja para monitorizar o estado dos vários recursos industriais identificando falhas ou problemas na produção.

Inicialmente esse controlo era quase inexistente e era predominantemente manual através de um operador humano, com o avanço da tecnologia o controlo passou a ser mais simplificado nomeadamente com o aparecimento do PLC (Programmable Logic Controller), tornando a deteção de uma falha mais simples e rápida pois o PLC pode facilmente acionar uma luz de controlo no painel de instrumentos do recurso.

No entanto hoje em dia existe a necessidade de que o controlo se processe o mais eficazmente possível, sendo requerido que esse controlo possa ser centralizado apenas num local e que esse local não seja dependente nem de um lugar físico nem de uma interface de comunicação específica, como é o caso de uma rede extensa de cabos dentro de um sistema de produção.

Observa-se então o surgimento de aplicações que possuem a capacidade de monitorizar e controlar remotamente os recursos, permitindo desta forma melhorias substanciais no controlo da produção bem como na manutenção dos recursos.

1.2 Enquadramento

A produção industrial tem sofrido grandes modificações e melhoramentos ao longo dos últimos anos possibilitando uma produção mais eficaz, eficiente e rápida. Essas melhorias devem-se em grande parte aos avanços nos campos da automação industrial e do controlo de processos.

Um sistema industrial é altamente heterogéneo e está constantemente sujeito a mudanças quer de produtos a produzir quer de novos recursos que necessitam de ser integrados no sistema [Filho, 2007]. Daí advêm problemas de integração essencialmente devido aos protocolos de comunicação distintos de cada recurso levando muitas vezes à obrigatoriedade de utilizar softwares ou implementações proprietárias de cada fornecedor de equipamento.

No sentido de facilitar essa integração e melhorar a flexibilidade e adaptabilidade dos sistemas de produção tem-se verificado nos últimos anos a adoção de uma arquitetura orientada a serviços (SOA, Service Oriented Architecture). O uso de serviços Web no sentido de implementar uma arquitetura SOA trouxe vantagens ao nível da comunicação entre processos em grande parte devido ao facto dos serviços Web assentarem na linguagem XML (Extensive Markup Language) o que os torna independentes do tipo de linguagem de cada recurso.

Os serviços Web permitem assim efetuar essa integração de uma forma simples e eficaz permitindo características tais como interoperabilidade, usabilidade e capacidade de reutilização.

Torna-se essencial hoje em dia integrar o nível superior e o nível inferior dos sistemas industriais, tal permite uma reação mais rápida a qualquer problema ou modificação no sistema de produção. A comunicação entre o nível primário da produção e os sistemas de controlo superiores como o ERP (Enterprise Resource Planning) e o MES (Manufacturing Execution Systems) é ainda muitas vezes manual e indireta, deste modo uma integração mais direta, viável e simples torna-se essencial no panorama industrial [Karnouskos, 2007].

A produção em grande escala bem como a elevada complexidade dos sistemas de produção tornou essencial em qualquer sistema de produção a existência de um controlo dos processos. Neste sentido surgiram sistemas chamados SCADA que são softwares que permitem a monitorização e o controlo dos processos.

Os processos industriais bem como muitos dos sistemas modernos dependem de sistemas SCADA para efetuar o controlo das suas complicadas operações. Exemplos típicos incluem controlo de redes elétricas, refinarias de petróleo, indústria farmacêutica e sistemas de abastecimento de água em que os sistemas SCADA efetuam operações de controlo e monitorização sobre uma estrutura industrial extremamente diversificada. O mercado do controlo e monitorização depende bastante da integração de sistemas e é esperado que cresça cerca de 300€ Biliões dos 188€ Biliões registados em 2007 para 500€ Biliões em 2020 [Karnouskos, 2011]. Tal irá apresentar um impacto significantes em vários domínios, nomeadamente na indústria automóvel, na indústria de processos bem como em outras áreas.

Com o avanço para uma infraestrutura que cada vez mais depende da monitorização do mundo real, da avaliação adequada da informação adquirida e da aplicação do controlo em tempo real leva a que os sistemas SCADA adquiram cada vez mais uma maior importância.

Os avanços nos sistemas industriais em conjunto com a evolução das tecnologias de informação, especificamente com o desenvolvimento da internet,

permitiram às empresas ter controlo do seu processo produtivo a partir de qualquer parte do mundo e em tempo real.

Os sistemas SCADA vêm-se hoje em dia obrigados a oferecer soluções que permitam aos seus utilizadores obter o nível de controlo desejado e ao mesmo tempo oferecer esse controlo e monitorização de uma forma remota e fácil de aceder, tomando partido das funcionalidades que a internet proporciona.

A arquitetura atual dos sistemas SCADA foi inicialmente desenvolvida tendo em mente o espaço singular de cada ambiente fabril, no entanto espera-se que haja potencial para aumentar as suas funcionalidades e ao mesmo tempo diminuir os custos de integração através da uma abordagem colaborativa a sistemas empresariais e serviços de grande escala a nível mundial. Os ambientes industriais tornam-se cada vez mais complexos deixando de ser viável (devido ao custo e ao tempo de desenvolvimento) criar sistemas individuais mas sim integrar SoS (System of Systems) de larga escala [Karnouskos, 2011].

Neste contexto a cooperação e interação entre uma aplicação SCADA e um conjunto de serviços Web que representem uma variedade de recursos existentes no ambiente fabril torna-se algo altamente desejável e uma grande mais-valia para qualquer processo industrial.

1.3 Objetivos

Pretende-se com este trabalho atingir os seguintes objetivos:

- Implementação de uma arquitetura orientada a serviços através de criação de serviços Web;
- Controlo e monitorização de um recurso industrial demonstrativo através desses serviços, permitindo que o recurso esteja disponível em qualquer parte do mundo através da internet;
- Criação de uma aplicação SCADA capaz de efetuar o controlo e a monitorização do recurso através da integração do serviço Web correspondente.

Pretende-se desta forma unir as mais-valias de uma solução SCADA e dos serviços Web para permitir o controlo e a monitorização de recursos industriais, independentemente da sua localização, da sua interface de comunicação, da plataforma informática usada e da linguagem de programação.

Em última análise pretende-se integrar os serviços Web e a tecnologia SCADA com o intuito de melhorar o controlo dos sistemas de produção e permitir que esse controlo esteja disponível vinte e quatro horas por dia e em qualquer sítio.

1.4 Organização da Dissertação

No capítulo 2 é apresentado o estado de arte e a revisão bibliográfica referentes aos temas relevantes ao contexto do presente trabalho. São apresentados os serviços Web bem como as tecnologias de suporte associadas aos mesmos, sendo ainda apresentada a tecnologia SCADA e a sua aplicação. São ainda analisados trabalhos científicos de outros autores que apresentam relevância para o enquadramento desta dissertação.

No capítulo 3 é apresentada a solução proposta.

No capítulo 4 é apresentada e descrita a implementação da solução proposta.

No capítulo 5 são apresentadas as considerações finais e trabalhos futuros.

2. Serviços Web e sistemas SCADA

2.1 Serviços Web

"Hoje em dia, a principal função da WWW (World Wide Web) é o acesso interativo a documentos e aplicações. Na maior parte dos casos, esse acesso é efetuado por utilizadores humanos, normalmente através de web browsers, dispositivos de reprodução de áudio, ou outro qualquer sistema interativo com essa capacidade. A web tem a potencialidade de crescer significativamente em poder e alcance se for estendida para suportar a comunicação entre aplicações, a partir de um programa para o outro."

- W3C XML Protocol Working Group Charter

Um serviço Web é um qualquer serviço que esteja disponível na internet, use um sistema de mensagens padrão XML e não seja dependente de nenhum sistema operativo ou linguagem de programação [Cerami, 2002]. É uma solução utilizada na integração de sistemas e na comunicação entre aplicações diferentes. Com esta tecnologia é possível que novas aplicações possam interagir com aquelas que já existem e que sistemas desenvolvidos em plataformas diferentes sejam compatíveis.

O consórcio W3C define serviço Web da seguinte forma: "Um Serviço Web é um sistema de software projetado para suportar uma interação entre máquinas interoperáveis numa rede. Tem uma interface descrita num formato processável por uma máquina, especificamente WSDL (Web Service Description Language). Outros sistemas interagem com o Serviço Web numa forma definida pela sua descrição através de mensagens SOAP (Simple Object Access Protocol), tipicamente através de HTML (HyperText Markup Language) e XML em conjunto com outros padrões Web" [Web Service, 2004].

Desta forma, um serviço Web completo é então qualquer um que [Cerami, 2002]:

- Se encontre disponível na internet ou numa rede privada (intranet);
- Use um sistema de mensagens padrão XML;
- Não esteja ligado a nenhum sistema operativo ou linguagem de programação;
- Se auto descreva através de uma descrição XML;
- Seja detetável através de um simples mecanismo de procura.

Visto que os serviços Web têm como objetivo principal a comunicação entre aplicações, ou seja o destinatário é um programa cliente e não um utilizador humano, a resposta fornecida por um serviço web deve ser passível de ser lida por uma máquina. Consequentemente deve obedecer aos protocolos que as máquinas (aplicações Web a correr em computadores) compreendem. Quando se escreve uma aplicação que apenas vai ser usada num ambiente limitado é possível decidir como esta vai operar. No caso dos serviços Web pretende-se que estes suportem uma comunicação entre aplicações e que o façam de uma forma uniforme, bem definida, aberta e com extensibilidade [Shklar, 2003].

É possível então analisar o conjunto de protocolos nos quais assenta a tecnologia dos serviços Web e que podem ser observados na figura 1 [Cerami, 2002]:

Transporte - Esta camada é responsável pelo transporte de mensagens entre aplicações. Inclui protocolos tais como HTTP (Hypertext Transfer Protocol), SMTP (Simple Mail Transfer Protocol), FTP (File Transfer Protocol) bem como protocolos mais recentes como o BEEP (Blocks Extensible Exchange Protocol).

Mensagens XML - Esta camada é responsável pela codificação das mensagens num formato XML o que torna possível a sua compreensão quer por parte do transmissor quer por parte do destinatário. Inclui presentemente mensagens SOAP.

Descrição - Esta camada é responsável pela descrição da interface de um determinado serviço Web. Essa descrição é efetuada através de um documento WSDL.

Descoberta - Esta camada é responsável pela centralização dos serviços num registo comum, possibilitando desta forma simples e eficazes funcionalidades de procura e publicação de serviços. A publicação é efetuada através do registo UDDI (Universal Description, Discovery and Integration).

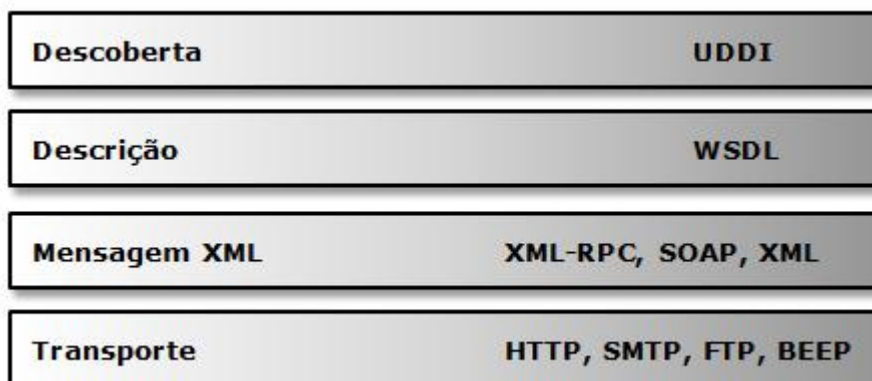


Figura 1 – Protocolos nos quais assentam os serviços Web [Cerami, 2002].

A figura 2 ilustra o processo de publicação, procura e acesso a um serviço Web, o processo pode ser explicado da seguinte maneira [Filho, 2007]:

1. **Registo**, este é o primeiro passo através do qual o fornecedor de serviços regista os seus serviços num registo UDDI.
2. **Pesquisa**, o cliente pode então procurar pelo serviço pretendido no registo UDDI.
3. **Descrição**, o cliente obtém a descrição dos vários métodos e tipos de parâmetros do fornecedor, isto é feito através da descrição WSDL, que identifica quais serviços estão disponíveis no serviço Web.
4. **Comunicação**, o cliente pode então invocar o serviço Web utilizando mensagens SOAP, que funcionam como um mediador capaz de interpretar o código XML presente nos pedidos e respostas trocados pelo cliente e o serviço Web.

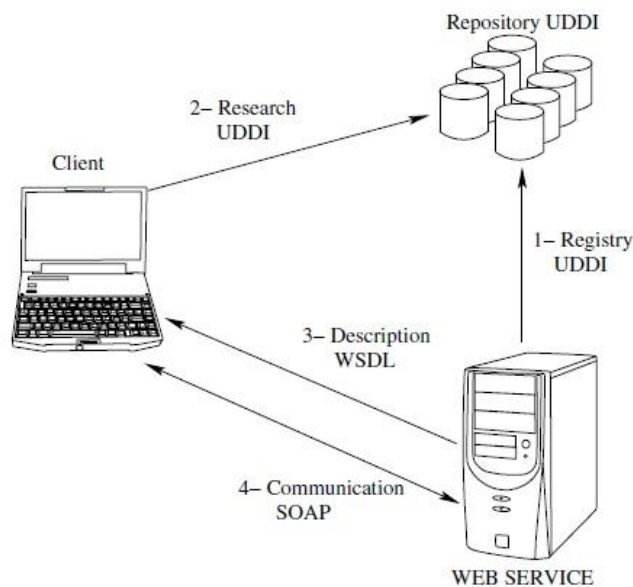


Figura 2 – Tecnologias que compõem um serviço Web [Filho, 2007].

2.2 Tecnologias de Suporte

2.2.1 SOAP

SOAP (Simple Object Access Protocol) é um protocolo que providencia um mecanismo simples e leve para a troca de informação estruturada entre pares num ambiente distribuído e descentralizado [SOAP 1.1], sendo essencial na troca de informação entre serviços Web e outras aplicações, figura 3.

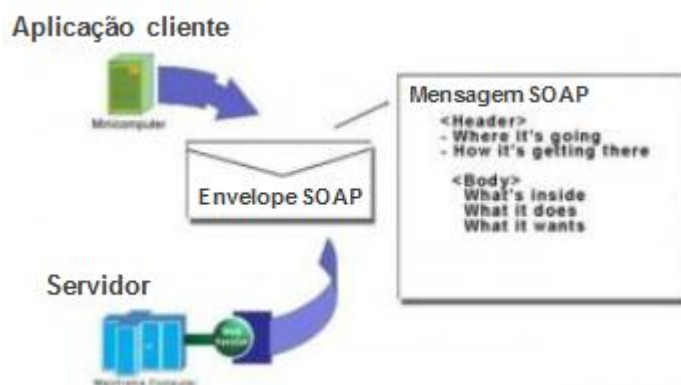


Figura 3 – SOAP na arquitetura dos serviços Web [WS-Tutorial, 2011].

SOAP define a sintaxe que permite o envio e recepção de mensagens XML entre o cliente e o servidor. Ou seja, SOAP é o “envelope” que contém as mensagens que vão ser enviadas através de HTTP entre os clientes e os serviços Web.

Fornece uma linguagem padrão que permite a interligação de aplicações e serviços. Uma aplicação envia um pedido SOAP para um serviço e o serviço Web envia a resposta através de uma resposta SOAP. SOAP pode potencialmente ser usado em combinação com uma variedade de protocolos, sendo que na prática é maioritariamente usado o protocolo HTTP. SOAP tem sido adotado como o padrão no que respeita aos serviços Web, desta forma aplicações de grandes corporações têm desenvolvido SOAP API's (Application Programming Interface, é uma especificação desenvolvida para ser usada como interface entre a comunicação de componentes dentro de um software) para os seus produtos facilitando em grande medida a integração de sistemas [Daconta, 2003].

Uma mensagem SOAP consiste num envelope que contém um cabeçalho opcional e um corpo como se pode observar na figura 4. O cabeçalho contém blocos de informação relevante ao modo como a mensagem é processada. Tal inclui configurações de mapeamento e entrega bem como afirmações de autenticação ou autorização e informações de comunicação. O corpo contém a mensagem propriamente dita, mensagem essa que vai ser processada e entregue, sendo que qualquer informação que possa ser expressa numa sintaxe XML pode ser incluída no corpo de uma mensagem [Tidwell, 2001].

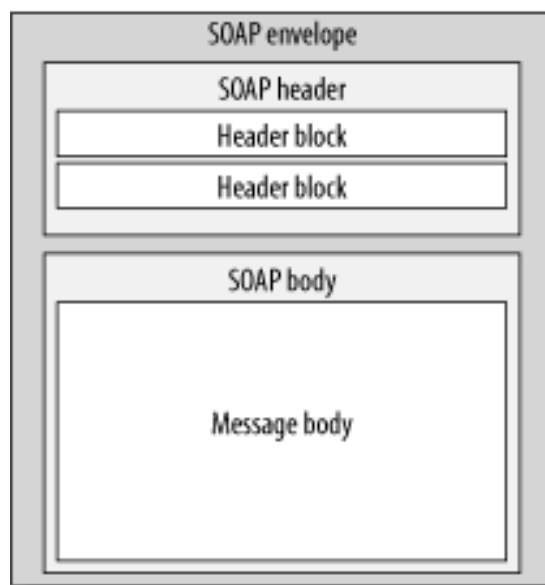


Figura 4 – Estrutura de uma mensagem SOAP [Tidwell, 2001].

```
<?xml version = "1.0" encoding = "UTF-8" ?>
<soap:Envelope xmlns:xsi = "http://www.w3.org/2001/XMLSchema-
instance"
xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
xmlns:soap = "http://schemas.xmlsoap.org/soap/envelope"
xmlns:soapenc
= "http://schemas.xmlsoap.org/soap/encoding">
<soap:Body>
<w:getWeather xmlns:w = "http://www.intlweather.com/services">
<degreeslong xsi:type = "xsd:float">-73.0</degreeslong>
<degreeslat xsi:type = "xsd:float">40.0</degreeslat>
<locale xsi:type = "xsd:string">en/us</locale>
</w:getWeather>
</soap:Body>
</soap:Envelope>
```

Figura 5 – Exemplo de um pedido SOAP com múltiplos parâmetros [Shklar, 2003].

Na figura 5 é possível observar um exemplo de um pedido SOAP com múltiplos parâmetros. O pedido é apresentado no seu formato XML. No exemplo o serviço Web retorna informação sobre o tempo que se faz sentir num determinado local. Deste modo é necessário definir a longitude e latitude bem como se se pretende obter informação no formato inglês ou americano (en/us), o que determina não só o tipo de linguagem mas também o tipo de medidas a apresentar referente ao sistema definido, Celsius ou Fahrenheit para a temperatura e milhas por hora ou quilómetros por hora para a velocidade do vento.

O envelope SOAP contém o corpo da mensagem, corpo esse que contém um único elemento que define a chamada remota de um procedimento, especificando o método pretendido (getWeather) e os seus argumentos (degreeelong, degreeelat e locale). O envelope é então transportado para um servidor SOAP através de um protocolo tal como HTTP ou SMTP.

Após receber o pedido, o servidor invoca o método especificado e gera a resposta que será transmitida à aplicação que a requereu. Na figura 6 é possível visualizar a resposta, que informa a aplicação cliente que a temperatura em Nova Iorque é de 25° (Fahrenheit), está parcialmente nublado e o vento sopra de sudoeste a uma velocidade de 5 milhas por hora.

O cliente SOAP pode então traduzir esta informação num formato que possa ser lido pelo utilizador humano. Tal pode ser efetuado usando uma das variadas SOAP API's disponíveis (por exemplo, Microsoft SOAP Toolkit ou JAXM/SAAJ) ou transformando o corpo da mensagem através de XSLT (Extensible Stylesheet Language Transformations) para um formato lido pelo utilizador humano (por exemplo, HTML, WML (Wireless Markup Language) ou VoiceXML) [Shklar, 2003].

```
<?xml version ="1.0" encoding ="UTF-8" ?>
<soap:Envelope xmlns:xsi ="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:xsd ="http://www.w3.org/2001/XMLSchema"
xmlns:soap ="http://schemas.xmlsoap.org/soap/envelope"
xmlns:soapenc
="http://schemas.xmlsoap.org/soap/encoding">
<soap:Body>
<w:getWeatherResponse xmlns:b
"http://www.intlweather.com/services">
<temperature xsi:type ="xsd:int">25</temperature>
<tempmode xsi:type ="xsd:string">F</tempmode>
<conditions xsi:type ="xsd:string">partly cloudy</conditions>
<windvelocity xsi:type ="xsd:int">5</windvelocity>
<windvelocityunits xsi:type ="xsd:string">MPH</windvelocityunits>
<winddirection xsi:type ="xsd:string">SE</winddirection>
</w:getWeatherResponse>
</soap:Body>
</soap:Envelope>
```

Figura 6 – Resposta ao pedido SOAP anterior [Shklar, 2003].

2.2.2 WSDL

O consórcio W3C define WSDL como “um formato XML para descrever serviços de rede como um conjunto de terminais que operam em mensagens que contenham informações orientadas a documentos ou a procedimentos” [WSDL 1.1]. De modo a que uma aplicação envie mensagens para um serviço Web particular, esta tem de analisar a descrição WSDL do mesmo e dinamicamente construir mensagens SOAP. Os

documentos WSDL descrevem então informação operacional: onde o serviço está localizado, o que faz o serviço e como falar com (ou invocar) o serviço.

WSDL (Web Service Definition Language) é o modo como se descreve os detalhes dessa comunicação e o tipo de mensagens que podem ser enviadas através do protocolo SOAP. É necessário referir que a linguagem WSDL, tal como o protocolo SOAP, baseia-se no formato XML. A linguagem WSDL pode considerada como um manual de instruções, na medida em que descreve de que maneira se pode utilizar um determinado serviço Web, por outro lado o programa cliente, consegue hoje em dia processar um documento WSDL e gerar automaticamente a mensagem SOAP necessária à invocação de um serviço específico.

O uso de WSDL apresenta várias vantagens, de entre as quais [Tidwell, 2001]:

- Torna possível uma fácil escrita e manutenção de serviços providenciando uma abordagem estruturada para a definição de interfaces para serviços Web.
- Reduz a quantidade de código necessário e os potenciais erros, tornando mais simples e eficiente a implementação de serviços Web por parte de uma aplicação cliente.
- Facilita a implementação de mudanças ao nível do código, mudanças essas que poderiam inviabilizar o bom funcionamento de aplicações cliente. A descoberta dinâmica de descrições WSDL permite que essas mudanças possam ser transmitidas automaticamente aos clientes usando WSDL de tal modo que as modificações ao código cliente não sejam necessárias cada vez que se efetua uma mudança no código.

Deste modo, a especificação WSDL usa os seguintes elementos na definição de serviços, [Cerami, 2002]:

- **definições** – Este elemento é o elemento base de todos os documentos WSDL. Define o nome do serviço Web, declara múltiplos “namespaces” usados por todo o documento e contém a descrição de todos os elementos do serviço.
- **tipos** – Este elemento descreve todos os tipos de dados usados entre o cliente e o servidor. Ainda que WSDL não use exclusivamente um esquema de linguagem, usa a especificação W3C XML Schema como a sua escolha por defeito.
- **mensagem** – Este elemento descreve as mensagens, quer seja uma mensagem de pedido ou uma mensagem de resposta. Define o nome da mensagem e contém informação referente aos parâmetros da mensagem e valores de retorno da mesma.

- **tipo de porta** – Este elemento define um serviço Web, as operações (métodos) que podem ser executadas, e as mensagens trocadas para executar a operação.
- **binding** – Este elemento define o mecanismo de transporte e o nome das operações que podem ser efetuadas usando o serviço Web.
- **serviço** – Este elemento define o endereço através do qual se pode invocar um serviço Web específico. Normalmente inclui um URL (Uniform Resource Locator) para a invocação do serviço SOAP.

Além destes seis elementos, a especificação WSDL define ainda mais dois elementos úteis, [Cerami, 2002]:

- **documentação** – Este elemento é usado para fornecer informação capaz de ser lida por utilizadores humanos e pode ser incluída dentro de qualquer dos outros elementos.
- **importar** – Este elemento é usado para importar outros documentos WSDL ou esquemas XML o que permite a criação de documentos WSDL mais modulares.

É importante observar que a linguagem WSDL não introduz uma nova definição de linguagem, mas reconhece a necessidade de sistemas mais elaborados para a descrição de mensagens e suporta a especificação XSD (XML Schemas Specification). No entanto como é irrealista pensar que apenas um tipo de linguagem será usado para descrever todos os formatos de mensagens, a linguagem WSDL permite o uso de outros tipos de linguagens através da extensibilidade.

O formato da linguagem WSDL pode possuir uma aparência bastante assustadora, no entanto não tem como objetivo ser lida por utilizadores humanos. Programadores e integradores não necessitam de entender WSDL ou SOAP para criar serviços Web, visto que quando se cria um serviço Web através de uma aplicação, a descrição WSDL é criada automaticamente pelo próprio software. A figura 7 mostra um exemplo de como esse processo funciona.

Na figura 7, o criador do serviço Web cria uma descrição WSDL através da ferramenta de desenvolvimento que analisa a camada de interface SOAP do serviço Web no passo 1. No passo 2, a aplicação cliente gera o código necessário para a comunicação com o serviço Web (cria um manipulador SOAP) através da análise da descrição WSDL. Finalmente no passo 3, a aplicação cliente e o serviço Web estão agora em condições de poderem comunicar um com o outro [Daconta, 2003].

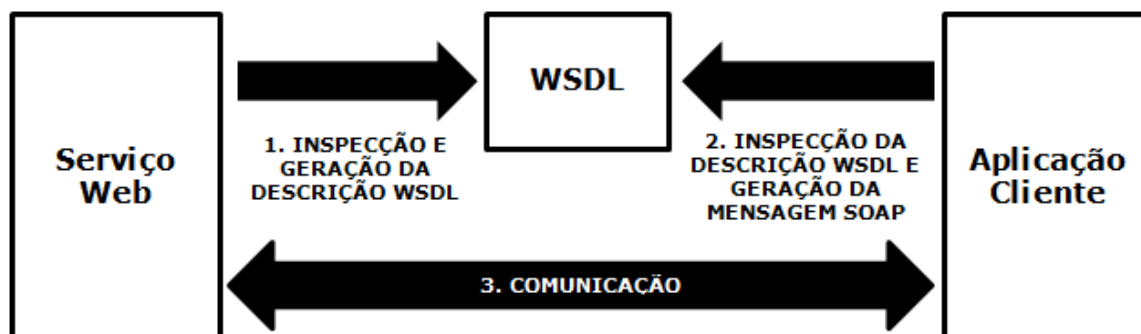


Figura 7 - WSDL como um mecanismo para facilitar a comunicação entre serviços Web e outras aplicações [Daconta, 2003].

2.2.3 UDDI

UDDI foi proposto em 2000 pelas empresas Ariba, Microsoft e IBM e foi criado para facilitar a descoberta de processos relacionados com serviços. Trata-se de uma iniciativa da indústria apoiada pela OASIS (Organization for the Advancement of Structured Information Standards), permitindo às empresas publicar os seus serviços e também descobrir os serviços de outras empresas, definindo como os serviços ou aplicações interagem na internet [Reckziegel, 2006].

O UDDI foi originalmente proposto como um mecanismo de apoio aos serviços Web, deste modo, foi desenvolvido para ser interrogado por mensagens SOAP e providenciar acesso a documentos WSDL (Web Services Description Language) que descrevem as vinculações do protocolo e o formato das mensagens requerido para interagir com os serviços listados no seu diretório [UDDI-W3C].

Hoje em dia o UDDI é uma tecnologia em evolução não sendo ainda considerada como uma tecnologia padrão, no entanto a implementação e desenvolvimento por parte de grandes empresas tem providenciado um grande avanço na sua conceção e melhoramento. Colocado de uma maneira mais simples o UDDI pode ser considerado como uma lista telefónica para serviços Web, as empresas podem registar informação pública sobre os seus serviços Web e sobre o tipo de serviços que disponibilizam no UDDI e por sua vez aplicações podem aceder ao registo e obter informação sobre os serviços disponíveis. Permitindo assim descobrir serviços Web tal como um computador descobre quantos dispositivos se encontram num dado momento na mesma rede de internet, [Daconta, 2003].

Como se pode verificar na figura 8, o registo UDDI é parte integrante de um serviço Web sendo um componente essencial na sua definição, permitindo a criação, especificação, descoberta e invocação de serviços Web.

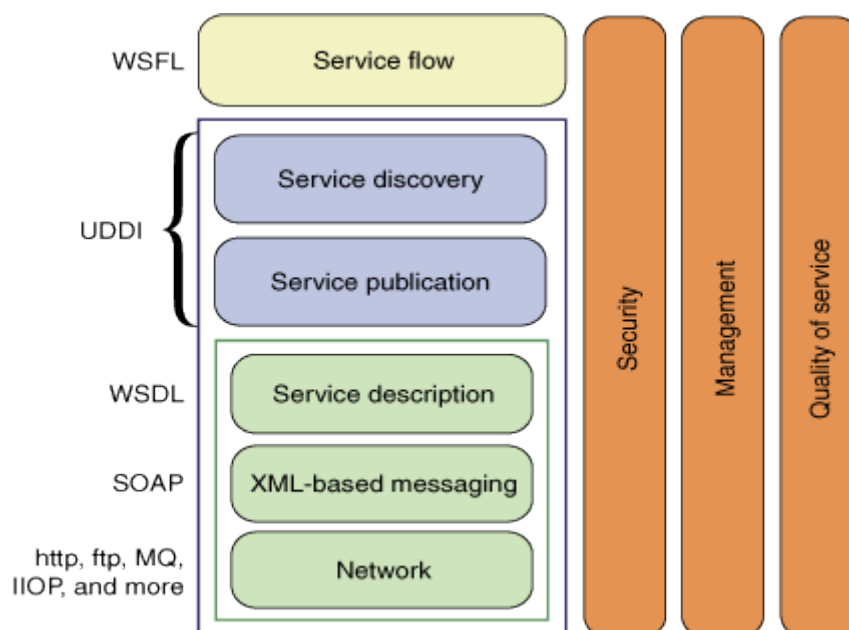


Figura 8 – As várias camadas de um serviço Web e o respetivo registo UDDI [Bellwood, 2002].

Para a integração interna dentro de uma empresa, o uso de um registo privado UDDI pode ser uma mais-valia. Dentro de uma grande organização onde várias aplicações de diferentes empresas necessitam de comunicar entre si, o uso de um registo privado dentro da organização poderá possibilitar essa comunicação entre serviços distintos. O uso desse registo, onde serviços são descritos através de WSDL, tem o potencial de minimizar o tráfego de documentação que é necessária existir hoje em dia entre empresas e corporações. Assim que as aplicações são capazes de produzir e ler mensagens SOAP, são descritas através de WSDL e têm a possibilidade de publicar um serviço no registo privado UDDI, passa a existir a possibilidade de projetos e aplicações dentro de uma mesma organização se poderem conectar dinamicamente, [Daconta, 2003].

O registo UDDI é constituído por três componentes ou páginas, na figura 9 é possível observar este conceito, [Wikipedia-UDDI]:

- **Páginas Brancas** – fornecem informação sobre a empresa que providencia determinado serviço. Isto inclui o nome da empresa e a sua descrição, potencialmente em diversas línguas. Usando esta informação é possível encontrar um serviço sobre qual se conhece de antemão algum aspeto do

mesmo, por exemplo localizar um serviço baseado no nome da empresa que o providencia. Informação de contacto para a referida empresa é também incluída, tal como uma morada ou um número de telefone ou email.

- **Páginas Amarelas** – fornecem uma classificação do serviço prestado por determinada empresa, baseado em determinadas normas definidas. Estas incluem SIC (Standard Industrial Classification), NAICS (North American Industry Classification System) ou UNSPSC (United Nations Standard Products and Services Code). O facto de cada empresa proporcionar vários serviços origina a existência de várias páginas amarelas (cada uma descrevendo um serviço) associadas a apenas uma página branca (que fornece informação genérica sobre a empresa).
- **Páginas Verdes** – São usadas para descrever como aceder a um serviço Web possuindo informação sobre as vinculações do serviço. Alguma da informação é relacionada com os serviços Web, caso do endereço do serviço, os seus parâmetros ou especificações. Outra parte da informação não se relaciona diretamente com os serviços Web, tal inclui informações relacionadas com emails, FTP, CORBA (Common Object Request Broker Architecture) e telefones referentes ao serviço em causa. Como um serviço Web possui múltiplas vinculações (vinculações essas definidas na descrição WSDL), um serviço poderá ter várias páginas verdes, pois cada vinculação terá de ser acedida de forma distinta.

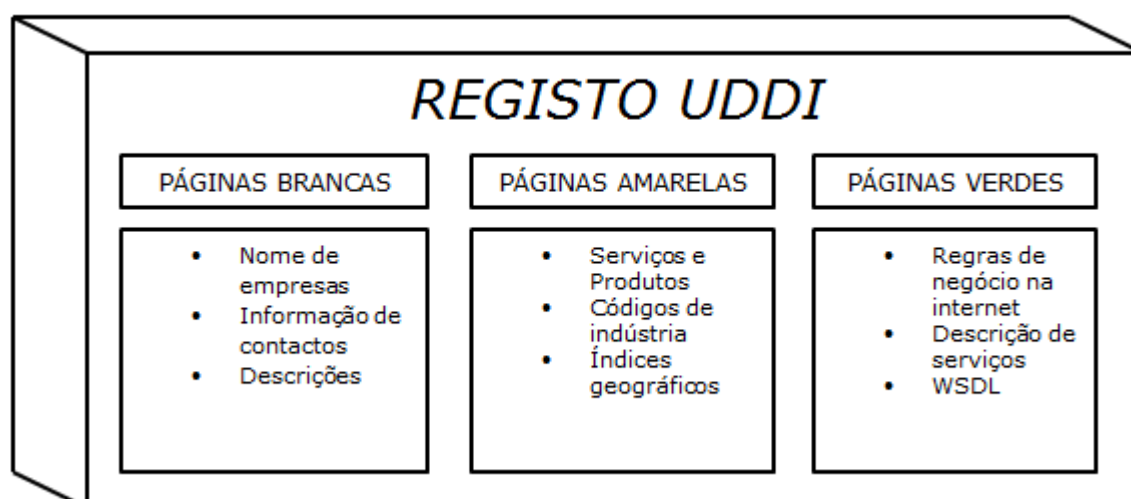


Figura 9 – Registo UDDI representado como uma lista telefónica [Daconta, 2003].

2.3 SCADA

SCADA (Supervisory Control And Data Acquisition) é um sistema que recebe informação de vários sensores numa instalação fabril, central de energia ou num outro local remoto e envia essa informação para um computador central que posteriormente efetua o controlo e gestão da informação.

Deste modo o termo SCADA é amplamente usado para representar soluções de controlo e supervisão numa grande variedade de indústrias, como é o caso dos sistemas de gestão de água e eletricidade de uma cidade, dos sinais de trânsito ou de qualquer sistema de produção de uma instalação fabril.

Quando se começou a efetuar a aquisição de dados era ainda usada uma lógica de relés no controlo da produção e dos sistemas fabris. Com o desenvolvimento do CPU (Central Processing Unit) e de outros equipamentos eletrónicos, os fabricantes começaram a incorporar o processamento digital no controlo através da lógica de relés, criando efetivamente o PLC, que é ainda hoje em dia um dos mais usados sistemas de controlo na indústria.

SCADA engloba a combinação de telemetria e aquisição de dados. SCADA efetua a recolha de informação através de unidades RTU (Remote Terminal Unit) como por exemplo PLC's, transmitindo a informação à central de análise, executa qualquer operação de análise e controlo necessária e exhibe a informação processada num número variado de ecrãs. As ações de controlo eventualmente necessárias são posteriormente transmitidas aos processos.

Com a crescente necessidade de monitorizar e controlar um número cada mais elevado de dispositivos no ambiente fabril, os PLC's tornaram-se mais distribuídos e os sistemas tornaram-se mais inteligentes e pequenos em tamanho. Desta forma generalizou-se a utilização de PLC's e/ou DCS's (Distributed Control System), sendo estes usados em rede como se pode visualizar na figura 10. Ainda que um RTU fosse inicialmente um dispositivo dedicado a uma tarefa específica, hoje em dia os PLC's são geralmente também usados como RTU's [Clarke, 2004].

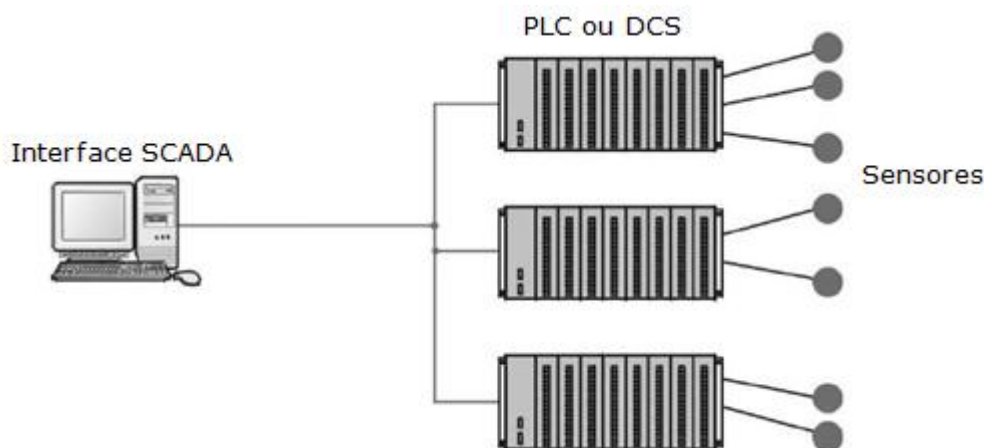


Figura 10 – Esquema de um sistema SCADA [Clarke, 2004].

As vantagens de um sistema SCADA são [Clarke, 2004]:

- A capacidade de o computador receber e guardar uma grande quantidade de informação;
- A informação pode ser exposta de qualquer maneira indo de encontro às necessidades do operador/cliente;
- Um grande número de sensores pode desta forma ser interligado numa mesma rede subordinada ao sistema de controlo;
- Vários tipos diferentes de dados podem ser lidos através dos RTU's;
- A informação recolhida pode ser visualizada em qualquer parte não se restringindo à consola HMI (Human-Machine Interface).

HMI's são uma forma simples de apresentar na mesma plataforma toda a informação recolhida dos RTU's. Torna-se difícil a monitorização individual dos mesmos visto que estes se encontram dispersos dentro do sistema de controlo e pelo facto de que estes não possuem usualmente uma forma de apresentar essa informação ao operador, desta forma o sistema SCADA comunica com os RTU's presentes em todo o sistema, processa a informação e consegue exibir essa informação facilmente numa interface HMI.

A informação recolhida por um sistema SCADA e apresentada numa interface HMI pode também ser guardada numa base de dados, base essa que pode usar a informação recolhida para elaborar gráficos, estatísticas e informação logística ou até ajudar a solucionar problemas. Na última década praticamente todos os sistemas SCADA passaram a integrar interfaces HMI e PLC's tornando possível executar e controlar mais facilmente um sistema SCADA [Tech-FAQ].

Razões que justificam a implementação de um sistema SCADA [Bailey, 2004]:

- Melhoria na produtividade da fábrica ou de um determinado processo devido à otimização do sistema;
- Aumento de produtividade dos operários;
- Melhoria da segurança do sistema derivada de uma melhor informação e de um controlo mais apertado;
- Maior eficácia na proteção dos equipamentos fabris;
- Proteção do ambiente fabril de uma possível falha no sistema;
- Melhoria na poupança de energia devido à otimização de todo o sistema;
- Melhoria e uma mais rápida receção de informação para que os clientes possam ser informados mais atempadamente e com mais precisão.

Os softwares SCADA podem ser divididos em dois tipos, proprietário e livre. Os fabricantes desenvolvem software proprietário para comunicar com o seu próprio hardware, desta forma, estes sistemas são vendidos como soluções ideais pois são perfeitamente compatíveis com o hardware, pois este pertence ao mesmo fabricante. O grande problema destes sistemas é a enorme dependência do fornecedor do sistema. Por outro lado os softwares livres têm ganho popularidade pois são capazes de interagir com qualquer dispositivo independentemente do fabricante o que proporciona ao sistema uma grande interoperabilidade. De entre os softwares existentes, utilizados hoje em dia, é possível referir alguns dos mais importantes: InTouch (Wonderware), RSview (Rockwell AB), Simplicity (General Electric), WinCC (Siemens), CitectSCADA (Schneider Electric) e Movicon (Progea).

Sendo este último utilizado no presente trabalho como plataforma de teste dos serviços Web utilizados na medida em que é um software SCADA sobre o qual havia uma maior familiarização no laboratório e pela sua disponibilidade.

2.3.1 Exemplo de um sistema SCADA

Um PLC pode controlar o fluxo da água de refrigeração dentro de um processo industrial, mas o sistema SCADA permite ao operador mudar as configurações desse fluxo permitindo a apresentação e registo de condições de alarme tais como a perda de fluxo ou temperatura elevada da água de refrigeração. Na figura 11 é possível observar um esquema do exemplo referido.

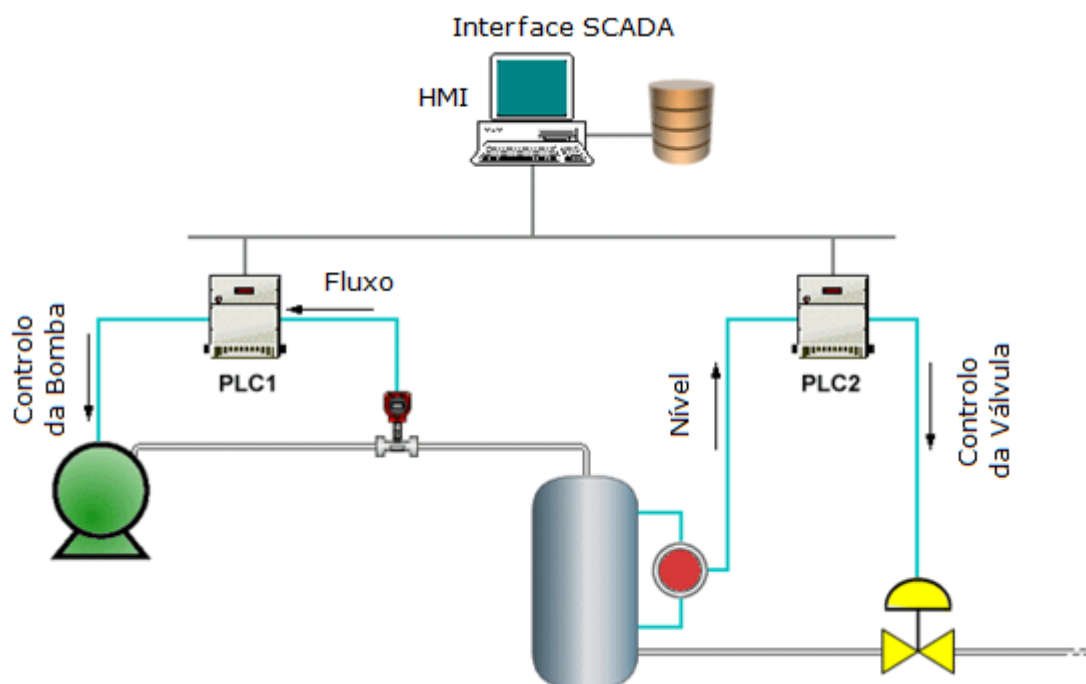


Figura 11 – Exemplo de um sistema SCADA [SCADA - TechnologyUK].

Para este caso a aquisição de informação começa ao nível dos PLC's e inclui a leitura de variáveis e o estado de equipamentos, sendo essa informação posteriormente comunicada ao sistema SCADA. Essa informação será posteriormente modificada e formatada de modo que possa ser apresentada ao operador através da interface HMI, possibilitando ao operador realizar operações que irão ajustar ou substituir as configurações prévias dos PLC's.

2.4 Alguns Trabalhos na Área

No contexto do desenvolvimento desta tese foram analisado vários trabalhos quer sejam eles artigos científicos ou dissertações de mestrado. Em seguida são apresentadas pequenas apresentações sobre os trabalhos analisados, referindo os seus objetivos e as suas conclusões:

“Integração de Sistemas de Produção” [Quintã, 2008]

A contínua evolução dos sistemas de produção, a globalização dos mercados e a competitividade levam a um nível de exigências cada vez mais elevado. A crescente inovação nas áreas do controlo de processos e da automação industrial têm levado a

que as ferramentas de comunicação entre processos assumam uma importância cada vez maior.

A revolução verificada na internet aliada à sua utilização de arquiteturas orientadas a serviços e a implementação de serviços Web levam à crescente importância das ferramentas baseadas na Web. A aplicação de arquiteturas orientada a serviços torna-se assim essencial para qualquer processo industrial, tendo como principal foco os serviços Web. No entanto apesar do tema ser já amplamente discutido e estudado faltam ainda aplicações reais com implementação na indústria.

Deste modo o autor propõe a utilização dessas mesmas ferramentas Web nos sistemas industriais, nomeadamente, propõe uma arquitetura orientada a serviços (SOA) como infraestrutura de suporte à integração de sistemas de produção distribuídos e descentralizados.

Foram analisados três casos de estudo nos quais se implementa a arquitetura SOA e onde são desenvolvidas plataformas baseadas em serviços Web. No caso de estudo 1 é abordado um recurso de transporte com controlo por PLC e no caso de estudo 2 é abordado um recurso de produção (centro de maquinaria) com controlador CNC (Computer Numerical Control). No caso de estudo 3 foi abordado o controlo integrado da produção (CIM, Computer Integrated Manufacturing) de um sistema flexível de produção (FMS, Flexible Manufacturing System), com vários recursos industriais de transporte e de produção, diversos tipos de controlo e diferentes protocolos de comunicação.

“Sistemas de supervisão e controlo integrados” [Costa, 2009]

O crescimento das economias e a sua necessidade constante de produtos transformou a produção em grande escala em algo comum e altamente automatizado nos dias de hoje. Isso tornou os autómatos programáveis (PLC) um elemento essencial de qualquer unidade industrial e a necessidade de controlar e supervisionar os processos criou sistemas chamados software SCADA que permitem a supervisão e o controlo em tempo real e local ou remotamente.

Os sistemas SCADA fazem sentido para instalações de grande porte, no entanto, para pequenas e médias empresas ou pequenas instalações com menor poder económico representam um investimento demasiado elevado.

Desta forma o autor refere a importância de aplicações “Freeware”, ou seja, o recurso a uma solução que permita a comunicação com os PLC’s e permita a leitura e controlo dos mesmos através da Web. A solução será então a criação de um sistema do tipo Cliente-Servidor, consistindo o sistema numa aplicação local capaz de

comunicar com os equipamentos e num conjunto de serviços Web que o cliente pode utilizar para efetuar o controlo de todos os processos.

O autor desenvolve um conjunto de serviços Web que permitem a um programador, mesmo sem conhecer a linguagem dos recursos, criar uma pequena aplicação SCADA que irá, remotamente, controlar e monitorizar um recurso ou mesmo criar uma estrutura para controlar e monitorizar um sistema flexível de produção, numa perspetiva de controlo integrado de produção.

“Web services-based automation for the control and monitoring of production systems” [Phaithoonbuathong, 2010]

A automação industrial tem ao longo dos anos fornecido grandes vantagens e inovações à produção industrial, no entanto, a complexidade da sua implementação e a dificuldade na sua modificação contínua para obedecer às tendências do mercado leva a uma perda de tempo na sua modificação e reconfiguração. Apresenta ainda uma dificuldade acrescida pois é necessário pessoal bastante especializado para a sua manutenção, tal obriga na maior parte dos casos à utilização de tecnologia específica de cada fornecedor o que resulta numa dificuldade acrescida para o cliente quando este pretende otimizar ou implementar um novo sistema de automação.

Os autores referem a importância da abordagem baseada em componentes (Component Based), no entanto com a progressão dos sistemas de produção tendo em vista uma produção mais rápida, mais eficaz e livre de erros torna-se necessário complementar essa abordagem com novos sistemas de automação e integração da produção que permitam um melhor controlo e monitorização de todo o sistema de produção. Nesse sentido a adoção de uma arquitetura orientada a serviços (SOA) baseada na implementação de serviços Web tem a capacidade de melhorar todos os aspetos dos sistemas de produção industrial.

Desta forma os autores implementam serviços Web numa célula de fabrico utilizando o protocolo DPWS (Device Profile for Web Services) para facilitar a comunicação e integração com os equipamentos de baixo nível, nomeadamente sensores e outros sistemas de controlo.

Os autores concluem que a implementação de serviços Web em sistemas de automação provou ser uma mais-valia para todo o processo de produção, permitindo uma integração e configuração mais simples de todo o controlo dos processos bem como a reconfiguração do sistema de controlo. A utilização de serviços Web permitiu a

evolução de todo o processo de automação providenciando uma capacidade de interoperabilidade entre vários dispositivos através de mensagens SOAP.

“Industrial Processes Supervision Using Service Oriented Architecture” [Filho, 2007]

A crescente evolução na automação industrial levou à existência de um número cada vez mais elevada de componentes num sistema industrial, desta forma a comunicação entre os componentes do sistema torna-se essencial e de grande importância para o processo industrial.

Softwares tais como CORBA ou OPC (OLE for Process Control) têm sido desenvolvidos com base numa arquitetura orientada a objetos, no entanto, a necessidade de aperfeiçoar a comunicação entre componentes e de melhores características como interoperabilidade ou flexibilidade, bem como facilitar a integração dos sistemas levou à adoção de uma arquitetura SOA orientada a serviços.

Esta arquitetura SOA assenta principalmente na utilização de serviços Web pois permitem uma mais simples integração de qualquer componente do sistema industrial essencialmente devido ao seu uso da especificação XML.

A implementação da arquitetura SOA é efetuada num caso de estudo em que os serviços Web assentam numa interface API Java Native Interface (JNI) e onde é utilizado um PLC SIMATIC S7-200 da Siemens para controlar o nível de um reservatório de água.

“Service-Oriented Device Communications using the Devices Profile for Web Services” [Jammes, 2007]

Para manter a competitividade exigida pela economia as empresas vêm-se obrigadas a adotar sistemas de produção que permitam uma rápida e simples reconfiguração bem como uma rápida e eficaz integração de qualquer novo componente necessário à produção de um novo produto.

Neste contexto a arquitetura SOA e o uso de serviços Web apresentam o potencial de providenciar a interoperabilidade necessária entre todos os componentes do sistema de automação industrial. A grande diversidade de componentes bem como os diferentes protocolos de comunicação levam ao isolamento dos componentes, desta forma, a utilização de um sistema de comunicação global tem a potencialidade de

eliminar a maioria dos problemas existentes ao nível da interoperabilidade e integração dos sistemas.

Tecnologias têm sido implementadas ao nível dos componentes nomeadamente UPnP (Universal Plug and Play) e Jini, no entanto, apresentam ainda obstáculos a uma integração ideal dos componentes, desta forma o protocolo DPWS apresenta-se como uma solução viável pois exhibe as mesmas vantagens do protocolo UPnP conseguindo uma integração perfeita com a tecnologia dos serviços Web.

O protocolo DPWS define uma arquitetura em que existem dois tipos de serviços: serviços hospedeiros e serviços que são hospedados. Serviços hospedeiros estão diretamente associados com um dispositivo e são parte fundamental no processo de descoberta desse dispositivo. Serviços hospedados são fundamentalmente funcionais e dependem do seu dispositivo hospedeiro para a sua descoberta. Além destes serviços hospedados o protocolo DPWS define ainda um conjunto de serviços:

- Serviços de descoberta – usados por cada dispositivo para se anunciar a ele próprio ou com o objetivo de descobrir outros serviços;
- Serviços de troca de metadados – providenciam acesso dinâmico à informação contida num serviço de um determinado dispositivo;
- Serviços associados a eventos – permitem que outros dispositivos possam subscrever mensagens de eventos produzidas por um determinado serviço.

No contexto do projeto ITEA (International Test and Evaluation Association)/SIRENA a Schneider Electric implementou a tecnologia DPWS em duas plataformas, tipo A (processador 44 MHz ARM7 TDMI com o software Thread X) e tipo B (processador 400 MHz Intel PXA 255 XScale com o software Windows CE) com vista à integração em qualquer componente para uma variedade de aplicações desde a automação industrial ao sector das comunicações.

“Integration of SOA-ready Networked Embedded Devices in Enterprise Systems via a Cross Layered Web Service Infrastructure”

[Karnouskos, 2007]

Torna-se essencial hoje em dia integrar o nível superior e o inferior dos sistemas industriais, tal permite uma reação mais rápida a qualquer problema ou modificação no sistema de produção. A comunicação entre o nível primário da produção e os sistemas de controlo superiores como o ERP e o MES é ainda muitas vezes manual e indireta, deste modo uma integração mais direta, viável e simples torna-se essencial no panorama industrial.

Os autores propõem a utilização de uma arquitetura orientada a serviços SOA assente em serviços Web, que utiliza o protocolo DPWS na comunicação e controlo dos componentes e um middleware proposto pelo projeto SOCRADES que faz a ponte entre os componentes e os sistemas como é o caso do ERP.

Tendo como objetivo principal a integração dos serviços de cada dispositivo com o software empresarial responsável pelo controlo e monitorização a arquitetura proposta apresenta quatro camadas:

- **“Device Layer”** – Todos os componentes do processo de produção e automação são representados nesta camada. Os dispositivos expõem a sua funcionalidade diretamente sob a forma de serviços Web através do protocolo DPWS;
- **“Composition Layer”** – Esta camada contém sistemas intermédios que fornecem serviços complexos através da utilização das capacidades dos dispositivos preparados para o protocolo DPWS. Esses serviços permitem realizar funções de controlo e gestão que um único dispositivo seria incapaz de realizar. Esta camada permite ainda a conexão de dispositivos, que não são capazes de fornecer serviços Web diretamente, à camada superior (“Middleware Layer”) através de controladores dedicados;
- **“Middleware Layer”** – Esta camada afeta a ligação entre as camadas inferiores e a camada superior (“Application Layer”). Possui uma subcamada que executa a invocação de serviços, o acesso aos dispositivos e trata dos eventos associados aos serviços. Possui outra subcamada que faz a ponte com a camada superior e trata da gestão do ciclo de vida de cada serviço e da administração dos dispositivos.
- **“Application Layer”** – Esta camada possui aplicações tradicionalmente ligadas ao nível superior como a supervisão e aplicações que até então não tiveram acesso às camadas inferiores como aplicações ERP ou sistemas de controlo como o MES. A exposição do serviço de cada dispositivo e o uso de protocolos de comunicação standard permitem que estas aplicações sejam capazes de aceder a qualquer dispositivo presente na rede com a mesma facilidade bem como permite que as aplicações possam invocar os serviços ou aceder diretamente os recursos de um determinado dispositivo.

A aplicação da arquitetura SOA permite uma integração mais simples, eficaz e livre de erros entre o nível inferior e superior de um sistema industrial de produção.

“SIRENA - Service Infrastructure for Real-time Embedded Networked Devices: A service oriented framework for different domains” [Bohn, 2006]

O projeto SIRENA foi iniciado em 2003 através de um consórcio de empresas do sector e com o apoio da Comissão Europeia inserido no programa ITEA com vista a desenvolver a arquitetura SOA com o objetivo de melhorar a integração de componentes em essencialmente quatro domínios distintos – automação industrial, sector das comunicações, produção automóvel e automação residencial.

Apesar de várias tecnologias terem sido estudadas no âmbito do projeto SIRENA, a tecnologia adotada foi o protocolo DPWS pois integra a funcionalidade dos serviços Web com vantagens sobre a tecnologia UPnP, possuindo serviços de descoberta, serviços de troca de metadados e serviços associados a eventos.

A plataforma “SIRENA Framework Enhancements” foi criada para facilitar o desenvolvimento, integração, manutenção e controlo dos dispositivos e serviços contidos numa interface SIRENA. Trata-se de um conjunto de ferramentas, componentes e serviços a seguir descritos:

- **DPWS-Stub-Skeleton-Generator** – Inclui ferramentas para a geração de “stub” (proxy) e esqueletos para a criação de serviços o que possibilita o acesso transparente às operações remotas de clientes;
- **WebWSDL** – Ferramenta desenvolvida pela Universidade de Rostock baseada na Web permitindo simplificar a definição da descrição WSDL de um dispositivo;
- **Device Lifecycle Management** – Serviço desenvolvido pela empresa Franhofer e usado para controlar o ciclo de vida do software de um dispositivo local tal como instalar, iniciar, parar, atualizar e remover dispositivos numa rede UPnP;
- **GINGER** – É um sistema de controlo de processos desenvolvido em Java pela empresa Kachel GmbH para a integração de aplicações e sistemas baseado em definições de fluxos de trabalho configuráveis;
- **Device Metadata Management** – Serviço criado pela empresa Capgemini em cooperação com a Schneider Electric tendo como objetivo estender as capacidades de gestão do protocolo DPWS através da requisição e mudança de informação de metadados referente a serviços remotos;

- **UPnP AV Stack** – É um software desenvolvido pela Universidade de Rostock que pode ser usado para o desenvolvimento de aplicação Audio/Video (MediaServer, MediaRenderer ou AV Control Points).

Desta forma foi implementada uma interface SIRENA com a finalidade de uma simples e mais eficaz integração e interação de componentes num contexto heterogéneo industrial. Na área da automação industrial criou-se um doseador industrial que transfere pequenos grânulos para pequenas garrafas e que contém vários sensores para monitorizar o processo e no contexto das comunicações criou-se um sistema que efetua o controlo de uma rede de telecomunicações, permitindo a configuração dinâmica bem como a deteção de falhas no equipamento.

“Integration of Automation Devices in Web Service supporting Systems” [Bangemann, 2009]

A evolução e o aparecimento de novas tecnologias criou um problema de integração com as tecnologias já existentes, a arquitetura SOA visa resolver esses mesmos problemas de integração utilizando serviços Web no entanto existem componentes que não estão preparados para uma implementação de serviços Web.

O projeto SOCRADES tem como objetivo resolver o problema de integração de componente que não possuem suporte para serviços Web no sistema industrial usando para o efeito um middleware, um “Mediator” que permite a ponte entre o protocolo DPWS associado à arquitetura SOA e o componente, especificamente a plataforma de comunicação PROFIBUS. É ainda utilizada a tecnologia OPC-UA para efetuar a comunicação entre o protocolo DPWS e a plataforma PROFIBUS.

Para o caso de estudo foi usada uma montagem de treino/demonstração da Festo Didactic desenvolvida na Universidade de Loughborough bem com um terminal elétrico CPX da Festo e um controlador com interface PROFIBUS. A solução apresentada oferece assim um grande potencial na adoção de aplicações baseadas em serviços Web devido à facilidade de integração dos componentes e permite uma gama bastante elevada de interfaces devido à utilização da tecnologia OPC-UA.

“A middleware for web service-enabled integration and interoperation of intelligent building systems” [Wang, 2007]

Hoje em dia o padrão IB (Intelligent Building) enfrenta dois grandes desafios: como ultrapassar as incompatibilidades bem como as oportunidades limitadas para a

integração de sistemas BAS (Building Automation Systems) numa variedade de produtos de múltiplos fabricantes; e como suplantar as dificuldades apresentadas na integração de sistemas BAS com a internet e com aplicações empresariais.

Os clientes deste tipo de tecnologia procuram dentro de um sistema BAS uma fonte de informação confiável e eficaz que facilite a gestão dos seus projetos. As exigências para uma boa integração da informação são hoje em dia muito mais amplas do que no passado, muito por parte da aceitação geral das novas tecnologias ethernet/internet/XML/serviços Web bem como a descida do preço das mesmas, o que as torna uma parte integrante da indústria IB.

Os autores apresentam o desenvolvimento de um middleware com o objetivo de realizar uma melhor integração e interoperabilidade entre os sistemas IB, assentando o middleware nas tecnologias OPC e serviços Web. O middleware utiliza a flexibilidade e a universalidade da tecnologia OPC para implementar a integração dos sistemas BAS numa rede LAN bem como diferentes tipos de comunicação são acomodados e convertidos numa linguagem unificada graças à interface OPC. Por outro lado os serviços Web, com a sua grande capacidade de comunicação numa plataforma heterogénea e na internet, permitem uma integração consistente do middleware e consequentemente do sistema BAS na internet.

Desta forma os métodos dos serviços Web fornecidos pelo middleware podem ser invocados por outras aplicações que efetuem o controlo dos sistemas presentes em edifícios, como é o caso de um sistema AVAC (Aquecimento Ventilação e Ar Condicionado) que pode otimizar o seu controlo através da informação obtida diretamente dos serviços Web do instituto meteorológico, da mesma forma aplicações empresariais como o ERP podem obter informação sobre vários parâmetros como a energia consumida no edifício.

“Interoperability of SCADA System Applications with Web Services” [Lipnickas, 2009]

Durante muito tempo houve problemas de comunicação entre equipamentos com formatação de dados incompatível o que trouxe problemas ao nível do funcionamento do sistema de controlo dos processos. O aparecimento da linguagem XML e das suas tecnologias adjacentes possibilitou a resolução desses problemas de incompatibilidade, proporcionando deste modo controlar o estado de um PLC particular dentro de um sistema e efetuar a monitorização de um processo remotamente num Web browser através das funcionalidades dos serviços Web.

Cada vez mais é exigido a um sistema de automação a capacidade de fornecer serviços tais como o controlo, a monitorização e a visualização da informação recolhida nos processos. A contínua evolução das redes de comunicação dentro de uma empresa significou que independentemente de como a informação é recolhida ou transmitida esta tem de estar disponível em qualquer parte de uma maneira simples e eficaz.

A crescente necessidade de um acesso remoto à informação favorece a visualização da mesma num Web browser ou em outro qualquer equipamento móvel em detrimento da informação fornecida localmente através de HMI's tradicionais.

A internet serve como uma plataforma que torna possível todo um conjunto de novas aplicações, nomeadamente aplicações de controlo remoto e distribuído ou sistemas SCADA de visualização e monitorização de informação, tal significa que os consumidores dessa informação têm ao seu dispor uma plataforma para a escrita e leitura de informação que se altera dinamicamente em qualquer parte do sistema industrial.

Desta forma os autores propõem uma arquitetura com três camadas que permita a passagem da informação entre vários PLC's dentro de uma rede e o cliente (neste caso através de um Web browser). Propõe-se então a visualização da informação recolhida de um conjunto de PLC's ligados numa rede ethernet. O software SCADA (Wonderware InTouch 9.5) recolhe essa informação e escreve-a numa base de dados (Microsoft SQL Server 2005) através do protocolo ODBC (Open Database Connectivity). Por sua vez o serviço Web comunica com a base de dados no formato XML e posteriormente comunica com o Web browser através de mensagens SOAP utilizando o protocolo HTTP, o Web browser por sua vez interpreta a informação em formato XML através da linguagem XSLT que permite gerar informação em formato HTML a partir do formato XML.

É assim possível verificar os benefícios da utilização do formato XML e a tecnologia dos serviços Web num sistema SCADA para atingir uma melhor integração de sistemas. Os serviços Web permitem uma simples e eficaz obtenção de informação sem que o cliente seja obrigado a aprender a comunicar com cada sistema devido ao XML que torna os serviços Web independentes da linguagem e da plataforma usadas.

“Architecting the next generation of service-based SCADA/DCS system of systems” [Karnouskos, 2011]

Os processos industriais bem como muitos dos sistemas modernos dependem de sistemas SCADA/DCS para efetuar o controlo das suas complicadas operações. Exemplos típicos incluem controlo de redes elétricas, refinarias de petróleo, indústria farmacêutica e sistemas de abastecimento de água em que os sistemas SCADA/DCS efetuam operações de controlo e monitorização sobre uma estrutura industrial extramente diversificada. O controlo e monitorização dependem bastante da integração de sistemas, tal irá ter um impacto significativo na indústria automóvel, indústrias de processos entre outras.

Com o avanço para uma infraestrutura que cada vez mais depende da monitorização do mundo real, da avaliação adequada da informação adquirida e da aplicação do controlo em tempo real vários problemas surgem.

A arquitetura atual dos sistemas SCADA/DCS foi inicialmente desenvolvida tendo em mente o espaço singular de cada ambiente fabril, no entanto espera-se que haja potencial para aumentar as suas funcionalidades e ao mesmo tempo diminuir os custos de integração através de uma abordagem colaborativa a sistemas empresariais e serviços de grande escala a nível mundial. Os ambientes industriais tornam-se cada vez mais complexos deixando de ser viável (tanto do lado do custo como do tempo de implementação) criar sistemas individuais mas sim integrar SoS de larga escala.

Os autores criam uma visão de como será a nova geração de sistemas SCADA/DCS identificando algumas mudanças fundamentais:

- Os sistemas são agora motivados pela informação e todas as interações são efetuadas através de serviços Web.
- Na perspetiva da arquitetura SOA todos os sistemas (ERP, PLC's, SCADA/DCS, MES, etc.) expõem as suas funcionalidades através de serviços que podem ser compostos e interagem com outras entidades.
- As interfaces HMI já não se encontram numa posição estática mas são agora acessíveis em qualquer lugar a qualquer hora através de dispositivos móveis.

Está-se perante o início de uma época onde complexos SoS vão afetar profundamente o mundo empresarial, tornando a monitorização e o controlo operações de elevada importância para qualquer aplicação do mundo real e dessa forma os sistemas e serviços envolvidos terão de ser capazes de lidar com as novas infraestruturas heterogéneas e de larga escala que se avizinham.

3. Solução

3.1 A crescente importância dos serviços Web

A criação de uma arquitetura orientada a serviços (SOA) trouxe consigo importantes contribuições no desenvolvimento de sistemas de controlo e na integração de recursos na produção industrial.

Os serviços Web têm nos últimos anos adquirido uma maior relevância no panorama industrial. Especialmente na integração de sistemas e no controlo de recursos devido à sua natureza independente da linguagem de programação e da plataforma informática, bem como à sua facilidade de implementação.

Os serviços Web permitem eliminar a dependência de redes de comunicação industriais pelo facto da sua comunicação se processar através de protocolos de internet e simplificam a integração pois podem ser implementados em qualquer sistema operativo e possibilitam que a aplicação de controlo seja ela também implementada em qualquer sistema operativo não comprometendo o sistema de controlo.

Além disso a sua capacidade de serem serviços prontos a consumir simplifica enormemente a reestruturação de um sistema de controlo e simplifica igualmente a sua incorporação em novos sistemas.

3.2 A importância de um sistema SCADA e as novas exigências do mercado

As soluções SCADA estão hoje em dia completamente incorporadas em certos sectores industriais e são indispensáveis no seu controlo, alguns exemplos incluem controlo de redes elétricas, refinarias de petróleo, indústria farmacêutica entre outros. Permitem uma simples e eficaz monitorização quer do nível primário de um ambiente fabril quer de um sistema de abastecimento de água, provando ser cruciais na centralização de informação bem como na sua difusão aos vários níveis de controlo de uma empresa.

A evolução nas tecnologias de informação bem como as exigências crescentes das empresas levaram a que a disponibilização da informação por parte de um sistema SCADA seja hoje em dia global e remota, ou seja, a informação deve estar disponível em qualquer parte do mundo a qualquer hora e não só a informação mas também o controlo dos sistemas quando aplicável deve ser possível.

Nesta perspetiva a integração de serviços Web num sistema SCADA permite eliminar qualquer preocupação ao nível da comunicação com os recursos nomeadamente o tipo de comunicação, o fabricante do recurso ou o software proprietário utilizado no recurso.

3.3 Solução encontrada

Como é referido nos objetivos pretende-se abordar e avaliar um novo tipo de solução SCADA, uma solução que se desvia dos protocolos de comunicação mais usados hoje em dia e passa a englobar os serviços Web como um meio de comunicação com os recursos num ambiente fabril.

Pretende-se desta forma introduzir uma abordagem nova ao controlo de sistemas de produção, combinando o melhor dos sistemas SCADA e as vantagens trazidas pela adoção de uma arquitetura SOA.

Procura-se então criar uma estrutura em que os recursos são controlados e monitorados através de serviços Web e por sua vez o sistema SCADA integra esses mesmo serviços centralizando toda a recolha de informação e o controlo dos equipamentos.

Desta forma pretende-se efetuar o controlo de um recurso industrial através da implementação de um serviço Web e posteriormente integrar o serviço Web numa solução SCADA.

Pretende-se implementar uma arquitetura com três componentes:

- **Recurso** – dispositivo industrial que tem como objetivo efetuar uma determinada tarefa dentro de um ambiente fabril, quer seja numa determinada etapa de uma linha de produção quer seja numa estação de trabalho singular. Linhas de transporte automatizadas, centros de maquinagem e manipuladores robóticos constituem alguns dos exemplos de um recurso no seio de um sistema de produção.
- **Serviço Web** – serviço que terá a responsabilidade de comunicar com o recurso, implementando o protocolo de comunicação necessário dependendo do dispositivo de controlo, sendo o PLC o mais habitualmente usado industrialmente. O serviço mediante os seus vários métodos deverá possuir a capacidade de obter o estado do recurso bem como enviar ordens ao mesmo para a execução de tarefas pré-determinadas.
- **Interface SCADA** – a aplicação cliente deverá ser capaz de exibir a informação respeitante ao funcionamento do recurso como deverá também implementar os controlos para que o recurso execute as tarefas necessárias ao

seu bom funcionamento. Integra o serviço Web, permitindo que as ordens dadas pelo utilizador ou programadas anteriormente na aplicação possam ser transmitidas ao recurso.

Obtém-se assim uma estrutura em três camadas que pretende simplificar, melhorar e tornar mais eficiente o controlo dos recursos num ambiente fabril. Na figura 12 é possível observar o conceito proposto, no qual um recurso é controlado e monitorado através de um ou mais serviços Web, por sua vez, o sistema SCADA recebe a informação dos serviços Web centralizando a visualização de informação e permitindo posteriormente difundir essa informação por todos os níveis do ambiente produtivo.

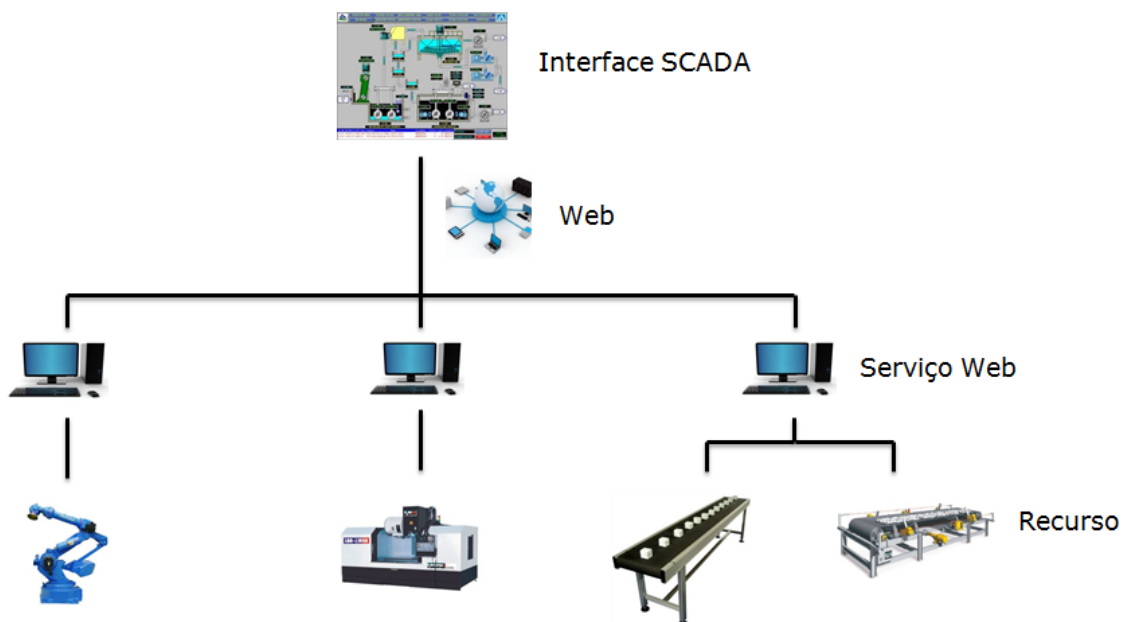


Figura 12 – Solução proposta.

4. Implementação

Tem-se então um recurso industrial, um tapete de transporte automatizado, que irá simbolizar um recurso num panorama fabril sendo o foco deste trabalho o controlo e monitorização do mesmo.

Para o efeito torna-se necessário seguir os seguintes passos para concretizar a implementação da solução:

- Compreensão dos equipamentos envolvidos no controlo do recurso bem como a elaboração de um programa ladder para o bom funcionamento do tapete;
- Efetuar a comunicação entre o PLC do recurso e o computador onde será implementado o serviço Web;
- Criação e implementação do serviço Web definindo os vários métodos necessários para o controlo do recurso;
- Desenvolvimento da aplicação cliente no software Movicon, integrando o serviço Web no mesmo.

É possível então observar a solução encontrada na figura 13:

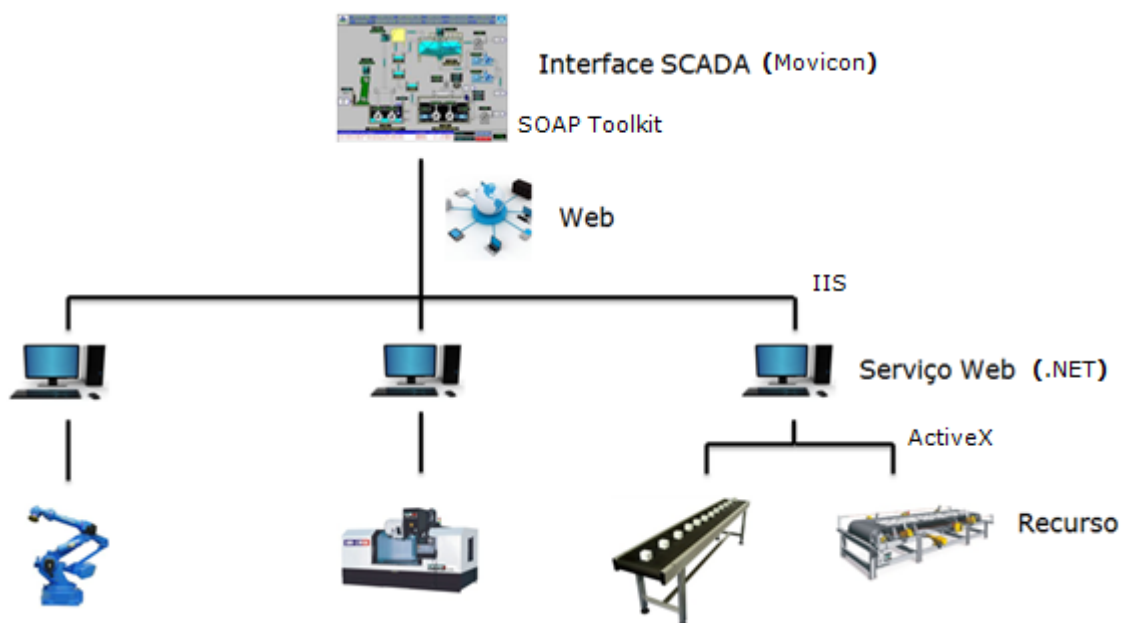


Figura 13 – Solução encontrada.

O serviço Web comunica com o recurso através de um controlo ActiveX, sendo criado através da plataforma .NET da Microsoft devido à sua facilidade e rapidez de criação de um serviço e também pela sua simples e direta implementação no IIS (Internet Information Services).

A solução SCADA foi elaborada através do software Movicon da Progea devido à sua disponibilidade e familiarização no laboratório. A comunicação entre o serviço Web e a solução SCADA é assegurada através do SOAP Toolkit v3.0 (apesar de já não ser suportado pela Microsoft esta foi a solução mais simples e eficaz encontrada para essa comunicação).

4.1 Recurso

O tapete de transporte presente no laboratório, figura 14, é constituído por vários componentes: contém um motor elétrico AC de 0,18 kW que realiza o movimento do mesmo, um variador de frequência NAIS VF-C Inverter, dois sensores fotoelétricos colocados em cada extremidade do tapete e um PLC Mitsubishi FX2N que efetua o controlo das saídas e entradas do sistema.

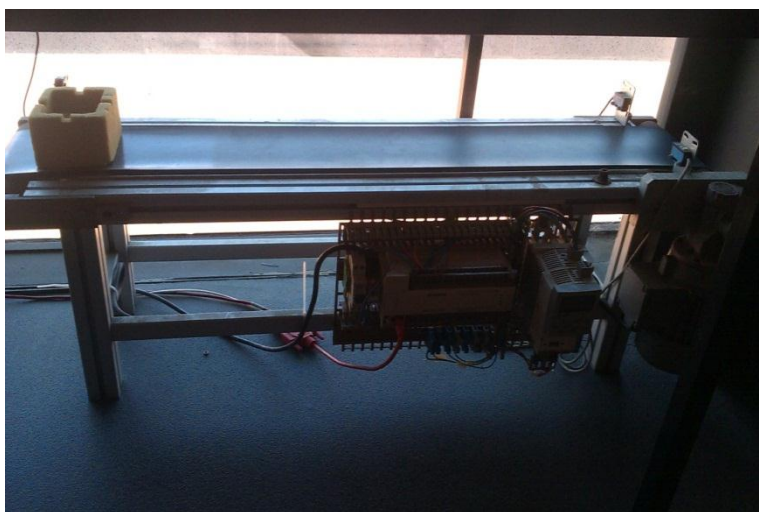


Figura 14 – Tapete de transporte presente no laboratório.

Para concretizar o objetivo de controlo o tapete e para evitar que o objeto a ser transportado não caia do mesmo é necessário ter em atenção algumas variáveis principais, sendo elas:

- **X1** – entrada digital que representa o sensor fotoelétrico da extremidade esquerda do tapete;

- **X2** – entrada digital que representa o sensor fotoelétrico da extremidade direita do tapete;
- **Y0** – saída digital responsável pelo acionamento do motor elétrico;
- **Y1** – saída digital que condiciona a direção do movimento do tapete (Y1=0 define o movimento para a direita e Y1=1 define o movimento para a esquerda);
- **M2** – variável de memória do PLC usada pelo serviço Web para a ativação do movimento do motor para a direita;
- **M3** – variável de memória do PLC usada pelo serviço Web para a ativação do movimento do motor para a esquerda;

Desta forma com a ajuda do software GX IEX Developer 7.01 foi elaborado um programa ladder relativamente simples que será posteriormente implementado no PLC do recurso. Este programa tem como objetivo ler os sensores, interromper o movimento do tapete quando o objeto atinge uma das extremidades e permitir o controlo do movimento através da manipulação das variáveis M2 e M3 por parte do serviço Web.

Deste modo a variável M2 quando ativada pela chamada do serviço ativa por sua vez a saída Y0 e desativa Y1 efetivamente colocando o tapete em movimento para a direita, sendo que a variável M3 ativa a saída Y0 e a saída Y1 colocando o tapete em movimento para a esquerda. É de notar ainda que desativando M2 e M3 interrompe-se o movimento do tapete.

No que respeita à comunicação entre o computador e o PLC esta é efetuada através do protocolo de comunicação RS232 sendo necessária a utilização de um conversor RS232/RS422 pois o PLC apenas possui uma porta de comunicação RS422. A interação entre o PLC e o computador é assegurada através de um controlo ActiveX que é configurado através do software MX Components v3.02C da Mitsubishi, que possibilita desta forma a implementação do protocolo de comunicação proprietário e fechado da Mitsubishi.

4.2 Serviço Web

O serviço "Tapete_Industrial" é criado através da plataforma .NET da Microsoft e tem como objetivo controlar o movimento do tapete e obter o seu estado.

Inicialmente para que a interação entre o serviço e o PLC seja possível torna-se necessário incluir uma biblioteca que contém as funções associadas ao controlo

ActiveX usado, funções necessárias para que as ordens possam ser transmitidas ao PLC.

Adiciona-se então uma referência à biblioteca "Mitsubishi ActMulti Controls V3.0" que implementa um conjunto de funções que são responsáveis pela leitura e escrita das variáveis do PLC, neste caso é apenas necessário utilizar duas dessas funções ReadDeviceRandom e WriteDeviceRandom respectivamente.

Torna-se então necessário definir os vários métodos associados ao serviço "Tapete_Industrial" para o controlo do tapete, visto que se trata apenas de um recurso demonstrativo definiram-se apenas quatro métodos simples, na figura 15 é possível observar a página Web com os métodos do serviço:

- **"tapete_direita"** – envia a ordem para o tapete se movimentar para a direita (M2=1), ativando as saídas digitais respetivas (Y1=0 e Y0=1);
- **"tapete_esquerda"** – envia a ordem para o tapete se movimentar para a esquerda (M3=1), ativando as saídas digitais respetivas (Y1=1 e Y0=1);
- **"tapete_paragem"** – envia a ordem para a paragem do tapete (M2 e M3=0), desativando a saída digital Y0;
- **"tapete_estado"** – obtém o estado do recurso, ou seja, o estado dos sensores (X1 e X2) e do motor (Y0 e Y1).

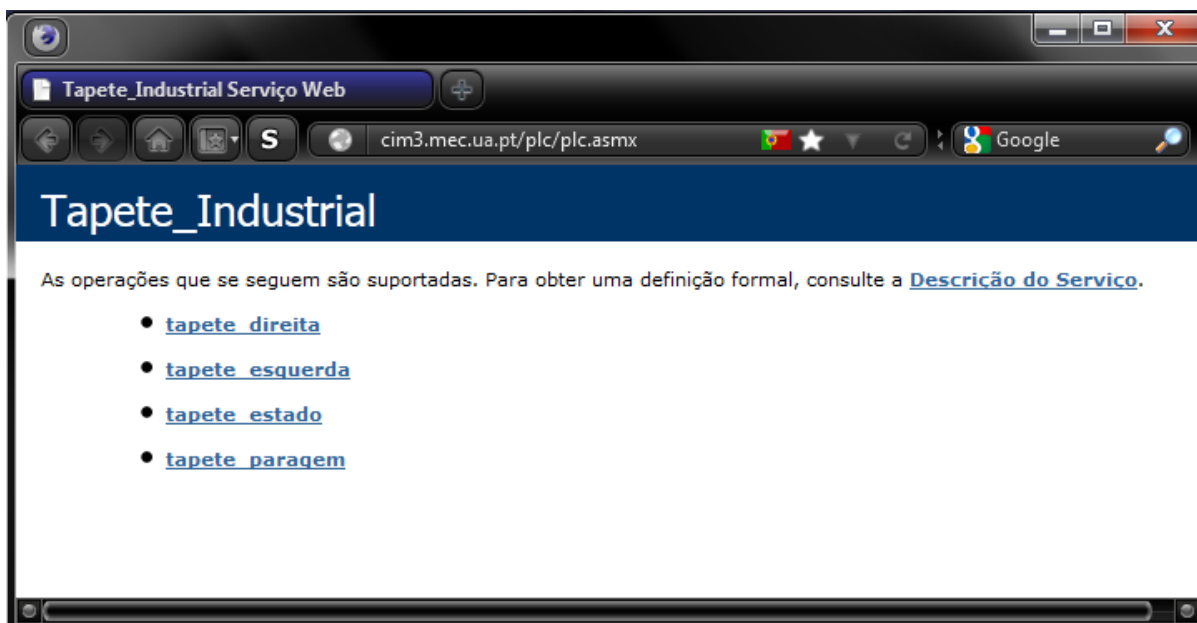


Figura 15 – Métodos do serviço Web.

Na figura 16 é possível observar o código necessário para a implementação do método "tapete_esquerda" no serviço "Tapete Industrial". É definido o PLC com o qual comunicar, neste caso o PLC tem o número 1022 atribuído, de seguida a ligação com o PLC é aberta e a definição do estado das variáveis é efetuada através da função WriteDeviceRandom e por fim a ligação com o PLC é terminada. É possível observar que a ordem de movimento para a esquerda é relativamente simples e fácil de implementar.

```

'#####
' _____ Método 2 _____
'Método para Mover o Tapete para a Esquerda (Left)
<WebMethod()> _
Public Function tapete_esquerda() As String
    mx.ActLogicalStationNumber = 1022
    mx.Open()

    'Mover Left ( Y1 = 1 )
    mx.WriteDeviceRandom("M2", 1, CLng(0))
    'Motor Ligado ( Y0 = 1)
    mx.WriteDeviceRandom("M3", 1, CLng(1))

    mx.Close()

    Return 0
End Function

```

Figura 16 – Código referente ao método "tapete_esquerda".

4.2.1 Implementação

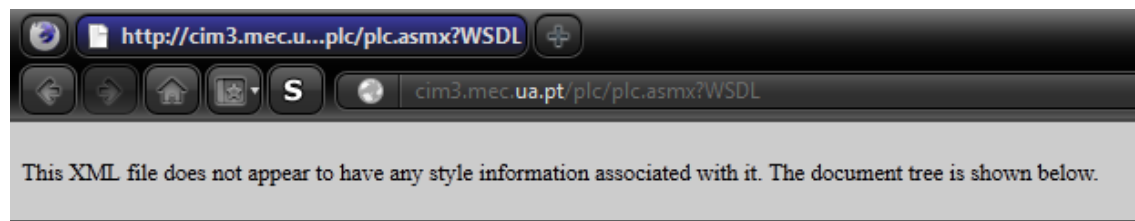
Tendo o serviço Web criado é necessário implementá-lo no IIS, ferramenta disponibilizada pelo sistema operativo, para que seja possível aceder e consumir o serviço através da internet tornando assim possível o controlo do tapete a partir de qualquer lugar.

É então necessário adicionar o serviço ao diretório virtual disponibilizado pelo IIS, tal é conseguido importando o aplicativo/ficheiro que contém toda a informação respeitante ao serviço. Cria-se desta forma um novo endereço no diretório virtual a partir do qual é possível aceder aos vários métodos do serviço. Neste caso o serviço é adicionado ao diretório virtual de um computador existente no laboratório com endereço definido <http://cim3.mec.ua.pt> em que o serviço publicado se encontra no seguinte endereço <http://cim3.mec.ua.pt/plc/plc.asmx>.

4.2.2 Descrição WSDL

A plataforma .Net na qual o serviço foi criado cria automaticamente uma descrição do mesmo, ou seja, cria um documento WSDL, no formato XML, que define as funcionalidades bem como os parâmetros e tipos de mensagens associados ao serviço.

A partir deste documento XML uma qualquer aplicação tem acesso à interface do serviço Web e pode através dela incorporar o serviço, para o serviço em questão a sua descrição WSDL está disponível em <http://cim3.mec.ua.pt/plc/plc.asmx?WSDL>, na figura 17 é possível visualizar uma versão simplificada dessa mesma descrição.



```

- <wsdl:definitions targetNamespace="http://cimteste.pt/">
  + <wsdl:types></wsdl:types>
  + <wsdl:message name="tapete_direitaSoapIn"></wsdl:message>
  + <wsdl:message name="tapete_direitaSoapOut"></wsdl:message>
  + <wsdl:message name="tapete_esquerdaSoapIn"></wsdl:message>
  + <wsdl:message name="tapete_esquerdaSoapOut"></wsdl:message>
  + <wsdl:message name="tapete_paragemSoapIn"></wsdl:message>
  + <wsdl:message name="tapete_paragemSoapOut"></wsdl:message>
  + <wsdl:message name="tapete_estadoSoapIn"></wsdl:message>
  + <wsdl:message name="tapete_estadoSoapOut"></wsdl:message>
  + <wsdl:message name="tapete_direitaHttpGetIn"/>
  + <wsdl:message name="tapete_direitaHttpGetOut"></wsdl:message>
  + <wsdl:message name="tapete_esquerdaHttpGetIn"/>
  + <wsdl:message name="tapete_esquerdaHttpGetOut"></wsdl:message>
  + <wsdl:message name="tapete_paragemHttpGetIn"/>
  + <wsdl:message name="tapete_paragemHttpGetOut"></wsdl:message>
  + <wsdl:message name="tapete_estadoHttpGetIn"/>
  + <wsdl:message name="tapete_estadoHttpGetOut"></wsdl:message>
  + <wsdl:message name="tapete_direitaHttpPostIn"/>
  + <wsdl:message name="tapete_direitaHttpPostOut"></wsdl:message>
  + <wsdl:message name="tapete_esquerdaHttpPostIn"/>
  + <wsdl:message name="tapete_esquerdaHttpPostOut"></wsdl:message>
  + <wsdl:message name="tapete_paragemHttpPostIn"/>
  + <wsdl:message name="tapete_paragemHttpPostOut"></wsdl:message>
  + <wsdl:message name="tapete_estadoHttpPostIn"/>
  + <wsdl:message name="tapete_estadoHttpPostOut"></wsdl:message>
  + <wsdl:portType name="Tapete_IndustrialSoap"></wsdl:portType>
  + <wsdl:portType name="Tapete_IndustrialHttpGet"></wsdl:portType>
  + <wsdl:portType name="Tapete_IndustrialHttpPost"></wsdl:portType>
  + <wsdl:binding name="Tapete_IndustrialSoap" type="tns:Tapete_IndustrialSoap"></wsdl:binding>
  + <wsdl:binding name="Tapete_IndustrialSoap12" type="tns:Tapete_IndustrialSoap"></wsdl:binding>
  + <wsdl:binding name="Tapete_IndustrialHttpGet" type="tns:Tapete_IndustrialHttpGet"></wsdl:binding>
  + <wsdl:binding name="Tapete_IndustrialHttpPost" type="tns:Tapete_IndustrialHttpPost"></wsdl:binding>
  + <wsdl:service name="Tapete_Industrial"></wsdl:service>
</wsdl:definitions>

```

Figura 17 – Descrição WSDL do serviço “Tapete_Industrial”.

4.2.3 Consumo

O consumo do serviço Web, ou seja, a utilização dos métodos que este disponibiliza, consiste na troca de mensagens do tipo pedido resposta em formato XML, de acordo com a especificação SOAP. Apesar do protocolo SOAP não definir um protocolo de transporte, na maioria dos casos práticos utiliza-se o protocolo HTTP [Quintã, 2008].

Desta forma quando a aplicação cliente invoca um dos métodos do serviço este irá enviar uma mensagem SOAP e receber da parte do servidor uma mensagem SOAP. Para o caso específico da invocação do método "tapete_esquerda" é possível analisar o pedido efetuado pelo cliente, figura 18, e a resposta obtida do servidor, figura 19, sendo que estas mensagens utilizam como método de transporte o protocolo HTTP.

```
1 POST /plc/plc.asmx HTTP/1.1
2 Host: cim3.mec.ua.pt
3 Content-Type: text/xml; charset=utf-8
4 Content-Length: 320
5 SOAPAction: "http://cimteste.pt/tapete_esquerda"
6
7 <?xml version="1.0" encoding="utf-8"?>
8 <soap:Envelope
9   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
10  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
11  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
12   <soap:Body>
13     <tapete_esquerda xmlns="http://cimteste.pt/" />
14   </soap:Body>
15 </soap:Envelope>
```

Figura 18 – Mensagem SOAP para invocar o método "tapete_esquerda".

Analisando a figura 18 é possível verificar que as primeiras cinco linhas referem-se à informação do protocolo HTTP, podendo identificar-se: a versão do mesmo, a informação que a mensagem é constituída por texto, a definição de que se trata de uma mensagem SOAP e define o URI (Uniform Resource identifier) a invocar. As restantes linhas compõem a mensagem propriamente dita ou seja o envelope contendo o corpo da mensagem (neste caso não existe cabeçalho) onde a parte mais importante é a indicação do método que deve ser executado, neste caso sendo o "tapete_esquerda".

```
1 HTTP/1.1 200 OK
2 Content-Type: text/xml; charset=utf-8
3 Content-Length: 350
4
5 <?xml version="1.0" encoding="utf-8"?>
6 <soap:Envelope
7   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
8   xmlns:xsd="http://www.w3.org/2001/XMLSchema
9   xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
10   <soap:Body>
11     <tapete_esquerdaResponse
12     xmlns="http://cimteste.pt/">
13       <tapete_esquerdaResult>0</tapete_esquerdaResult>
14     </tapete_esquerdaResponse>
```

Figura 19 – Mensagem SOAP de resposta ao método "tapete_esquerda".

No caso da figura 19 a estrutura é em todo semelhante ao pedido, as três primeiras linhas correspondem à informação respeitante ao protocolo HTTP, no entanto neste caso a resposta é enviada através do método GET ao contrário do pedido que utiliza o método POST. Neste caso o corpo da mensagem SOAP não apresenta informação de grande relevância pois este método não retorna nenhum valor diretamente pedido pelo método, retorna apenas "0" caso a invocação do método tenha tido sucesso podendo desta forma ser de algum interesse dependendo do programador ou da aplicação na qual o serviço é integrado.

4.3 Aplicação SCADA

Estando o serviço criado e publicado resta agora desenvolver a aplicação cliente, "Tapete_Cliente_Movicon", que irá integrar o serviço Web e permitir a monitorização e o controlo do tapete de transporte automatizado.

Visto que se trata do controlo de um recurso demonstrativo a ênfase da solução SCADA está centrada no desenvolvimento de uma interface gráfica de fácil leitura e numa estrutura simples que permita o controlo do tapete.

Desenvolve-se então uma interface, observada na figura 20, que de uma maneira concisa e fácil de interpretar indica se o tapete se encontra em movimento, indica a direção segundo a qual o movimento é efetuado e indica quando o objeto a transportar se encontra numa das extremidades do tapete.

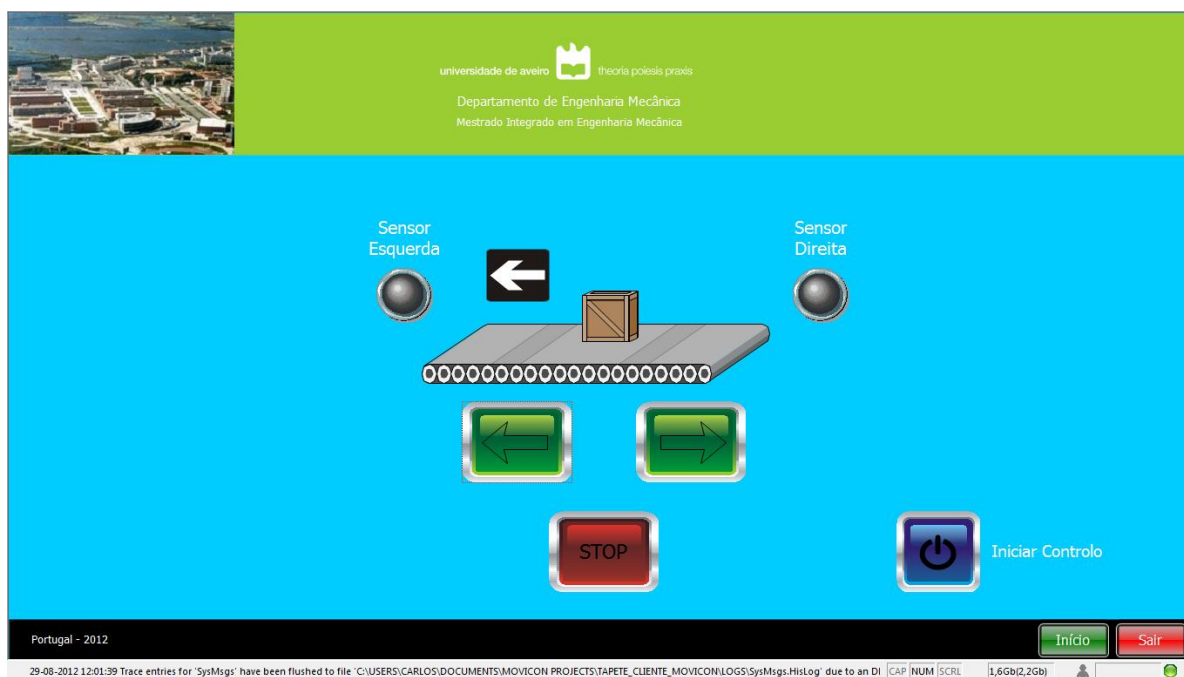


Figura 20 – Interface de controlo da aplicação SCADA.

Para o efeito faz-se uso de vários itens do interface, nomeadamente, os 3 botões centrais para controlar o movimento e posição do objeto no tapete, 2 luzes que indicam se o objeto alcançou uma das extremidades do tapete, 2 imagens que indicam em que direção o tapete se move e um botão azul que inicia o controlo do recurso

A interface entre o serviço Web e a aplicação SCADA é possível graças à biblioteca de funções "Soap type library v3.0" da Microsoft que implementa as funções necessárias para o envio e receção de mensagens SOAP. Graças à descrição WSDL do serviço Web é então possível efetuar o seu consumo dentro da aplicação SCADA visto que a biblioteca de funções adicionada permite estabelecer a troca de mensagens SOAP entre a aplicação SCADA e o serviço Web.

4.3.1 Integração do serviço Web

Após a instalação da biblioteca de funções no sistema operativo e tendo a interface SCADA projetada torna-se necessário atribuir ações a cada um dos itens presentes na interface para que o controlo e a monitorização sejam possíveis.

Usando o botão que permite movimentar o tapete para a esquerda como exemplo e atendendo a que o software MOVICON suporta VBA (Visual Basic for Applications) para a programação dos itens torna-se inicialmente necessário incluir a biblioteca ao item específico, figura 21.

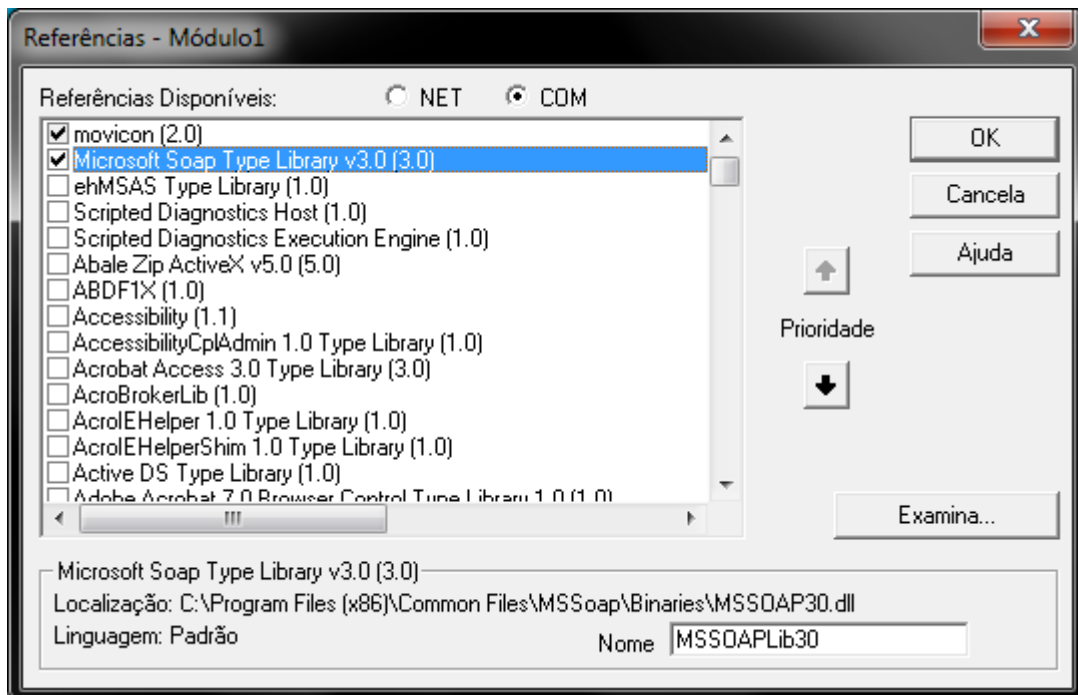


Figura 21 – Inclusão da biblioteca de funções.

A partir deste momento tem-se acesso às funções que permitem efetuar a comunicação com o serviço Web. Na figura 22 é possível observar o código para esse mesmo botão da interface gráfica da solução SCADA. Através da análise do código é possível constatar que o primeiro passo consiste na criação de um cliente SOAP, de seguida tendo acesso à descrição WSDL do serviço pretendido é possível efetuar a iniciação do cliente SOAP identificado no exemplo por "tapete1". A partir deste momento o cliente SOAP está preparado para invocar diretamente qualquer um dos quatro métodos disponíveis para o serviço "Tapete_Industrial". Neste caso é invocado o método "tapete_esquerda" pois pretende-se movimentar o tapete para a esquerda.

```
'#Language "WVB.NET"
Imports System.Exception
Imports System.IO

Option Explicit

Public Sub Click()

    'Criação do cliente SOAP
    Dim tapete1 As New MSSOAPLib30.SoapClient30

    'Iniciação do cliente SOAP através da descrição WSDL do serviço
    tapete1.MSSoapInit("http://cim3.mec.ua.pt/plc/plc.asmx?WSDL")

    'Invocação do método "tapete_esquerda" que movimenta o tapete para a esquerda
    tapete1.tapete_esquerda()

End Sub
```

Figura 22 – Código do botão que movimenta o tapete para a esquerda.

No caso da monitorização do recurso torna-se necessária a utilização de um "timer" que periodicamente efetua o pedido do estado do recurso. O método "tapete_estado" tem por objetivo retornar o estado do recurso e fá-lo através do retorno de uma string com quatro caracteres, cada caracter representando o estado (ON=1, OFF=0) das variáveis X1, X2, Y0 e Y1, partindo desta informação rotinas de comparação são capazes de identificar o estado do recurso que pode ser observado na interface gráfica da figura 19. Para o efeito criam-se variáveis que serão associadas ao estado de cada item, o estado das variáveis são definidos pelo resultado do método "tapete_estado" e por sua vez as variáveis condicionam o aparecimento das imagens que indicam a direção de movimento e as luzes que sinalizam as extremidades do tapete. O código responsável pela análise da informação retornada pelo método "tapete_estado" pode ser visualizado na figura 23.

A utilização deste timer encontra-se condicionada por um botão que deve ser premido para iniciar o controlo do recurso, este botão permite desta forma efetuar o arranque do timer que a cada tique verifica o estado do recurso.


```
'Sensor Esquerda
If x1 = "1" Then
    SetVariableValue("sensor_left", 1)
Else
    SetVariableValue("sensor_left", 0)
End If

'Sensor Direita
If x2 = "1" Then
    SetVariableValue("sensor_right", 1)
Else
    SetVariableValue("sensor_right", 0)
End If

'Movimento Motor    y1=1 --> Esquerda    y1=0 --> Direita
If y0 = "1" Then
    If y1 = "1" Then
        SetVariableValue("move_left", 1)
    Else
        SetVariableValue("move_right", 1)
    End If
Else
    SetVariableValue("move_left", 0)
    SetVariableValue("move_right", 0)
End If
```

Figura 23 – Rotina de comparação que possibilita a monitorização do recurso.

5. Considerações Finais

No presente trabalho foi avaliada uma nova abordagem aos sistemas de controlo de produção, nomeadamente apresentando uma nova solução SCADA no que toca ao controlo e monitorização de recursos industriais.

No capítulo 1 foi apresentado o problema indicando a necessidade crescente de um maior e mais complexo controlo do ambiente fabril, a adaptação dos sistemas SCADA às novas tecnologias e às novas exigências do mercado, a crescente importância dos serviços Web no contexto da arquitetura SOA e o problema da integração de recursos após a modificação ou reconfiguração de uma linha de produção. Referir ainda a necessidade de se obter hoje em dia um sistema de controlo cada vez mais flexível e descentralizado que possua a capacidade de estar disponível 24 horas por dia e em qualquer parte do globo.

No capítulo 2 identificaram-se as principais características das tecnologias dos serviços Web e dos sistemas SCADA, analisando ainda as tecnologias de suporte de um serviço Web. Foram também estudados vários trabalhos científicos relevantes para a dissertação em causa quer seja na temática dos serviços Web, dos sistemas SCADA ou dos sistemas de controlo industriais.

No capítulo 3 foi apresentada a solução proposta sendo esta constituída por uma estrutura com três níveis em que um recurso comunica com um serviço Web e o serviço Web comunica com uma aplicação SCADA. Esta aplicação possibilita então a difusão da informação e o controlo da produção tornando-os acessíveis a qualquer nível de um ambiente industrial.

No capítulo 4 foi abordada a implementação da solução SCADA referindo os três níveis da plataforma, nomeadamente:

- A identificação, configuração e comunicação com o recurso a controlar/monitorar;
- O desenvolvimento do serviço Web, a sua comunicação com o recurso e a sua posterior publicação numa plataforma (IIS) tornando o serviço acessível de qualquer parte;
- O desenvolvimento da interface gráfica da aplicação cliente (SCADA) e a integração do serviço Web possibilitando o controlo dos processos.

Desta forma é possível verificar a facilidade de criação e implementação dos serviços Web bem como o seu potencial na crescente interoperabilidade entre os recursos e os sistemas de controlo, bem como a sua independência face ao tipo de

linguagem de programação usada. De notar que nesta abordagem o código necessário para a comunicação com o recurso é implementado pelo serviço não se fazendo uso de uma aplicação local como interface entre os dois, tornando necessário personalizar o serviço para cada recurso consoante o seu protocolo de comunicação.

No que se refere à aplicação SCADA, esta passa de uma plataforma de comunicação restrita aos protocolos proprietários que reconhece para uma plataforma capaz de ignorar esse aspeto devido à simplicidade de integração e comunicação com um serviço Web.

Desta forma elimina-se também qualquer dependência de cabos ou protocolos de comunicação físicos, permitindo à aplicação SCADA ser implementada e configurada em qualquer sítio sem o constrangimento do espaço físico de uma unidade de produção.

Torna-se assim mais intuitiva a integração de um novo recurso fabril bastando para tal a modificação do serviço Web para acomodar a comunicação com o recurso. Tendo o serviço pronto a comunicar com o recurso este pode ser integrado direta e facilmente na aplicação de controlo permitindo desta forma uma rápida integração no processo produtivo.

Esta abordagem foi efetuada com um ponto de vista académico apresentando as suas limitações e como tal existiam ainda alguns aspetos que convém ter em conta, nomeadamente, a existência de alterações quando da utilização de vários recursos e/ou do seu controlo simultâneo, a capacidade dos vários softwares SCADA de integrar o serviço Web e os aspetos de segurança inerentes à utilização da internet como plataforma de comunicação.

5.1 Trabalhos Futuros

No que respeita a trabalhos futuros existem ainda várias etapas que convém salientar e estudar:

- A implementação da tecnologia OPC na medida em que facilita a comunicação entre os recursos e os serviços Web;
- A implementação de serviços Web na plataforma Java e a utilização de serviços Web RESTful;
- A integração da solução num ambiente industrial;
- A integração da informação obtida através do cliente SCADA num sistema de controlo superior ERP.

Bibliografia

[Bailey, 2003]

Bailey, David - *Practical SCADA for Industry*. Newnes, Elsevier, 2003, 288 páginas.
ISBN - 07506 58053

[Bangemann, 2009]

Bangemann, Thomas [et al.] - *Integration of Automation Devices in Web Service supporting Systems*. Preprints of the 30th IFAC Workshop on Real-Time Programming and 4th International Workshop on Real-Time Software. P. 161-166, 2009, [citado em 01-03-2012].

Disponível em: <<http://www.socrades.eu/Documents/objects/file1259605272.78>>.

[Bellwood, 2002]

Bellwood, Tom. - *Understanding UDDI*. 1 Julho 2002. [citado em 28-09-2011].
Disponível em: <<http://www.ibm.com/developerworks/webservices/library/ws-featuddi/>>.

[Bohn, 2006]

Bohn, H. [et al.] - *SIRENA - Service Infrastructure for Real-time Embedded Networked Devices: A service oriented framework for different domains*. Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies, 2006, ICN/ICONS/MCL 2006. P. 43, 8 Maio 2006, [citado em 01-03-2012].

Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1628289>>.

ISBN 0-7695-2552-0.

[Cerami, 2002]

Cerami, Ethan - *Web Services Essentials*. 1.º Edição. O'Reilly, Fevereiro 2002. 304 páginas.

ISBN 0-596-00224-6.

[Clarke, 2004]

Clarke, Gordon [et al.] - *Practical Modern SCADA Protocols: DNP3, 60870.5 and Related Systems*. Newnes, Elsevier, 2004, 536 páginas.

ISBN 07506 7995

[Costa, 2009]

Costa, Diogo da Silva - *Sistemas de supervisão e controlo integrados*. Aveiro, Universidade de Aveiro, 2009. Tese de dissertação apresentada à Universidade de Aveiro para a obtenção do grau de Mestre em Engenharia Mecânica.

[Daconta, 2003]

Daconta, Michael C.; Obrst, Leo J.; Smith, Kevin T. - *The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management*. John Wiley & Sons Ltd, 2003. 281 páginas. ISBN - 0-471-43257-1.

[Filho, 2007]

Filho, F. S. L. [et al.] - *Industrial Processes Supervision Using Service Oriented Architecture*. IEEE Industrial Electronics Society - 32nd Annual Conference. P. 4552-4556, 16 Abril 2007, Paris, França, [citado em 01-03-2012].

Disponível

em:

<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4153488>>.

ISBN 1-4244-0390-1.

[Jammes, 2007]

Jammes, François [et al.] - *Service-Oriented Device Communications using the Devices Profile for Web Services*. Advanced Information Networking and Applications Workshops, 2007, AINAW '07 - 21st International Conference. P. 947-955, 27 Agosto 2007, Grenoble, França, [citado em 01-03-2012].

Disponível

em:

<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4221180>>.

ISBN 978-0-7695-2847-2.

[Karnouskos, 2007]

Karnouskos, S. [et al.] - *Integration of SOA-ready networked embedded devices in enterprise systems via a cross-layered web service infrastructure*. *Emerging Technologies and Factory Automation, 2007, ETFA, IEEE Conference*. P. 293-300, 4 Janeiro 2008, Patras, Grécia, [citado em 01-03-2012].

Disponível

em:

<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4416781>>.

ISBN 978-1-4244-0825-2.

[Karnouskos, 2011]

Karnouskos, Stamatis [et al.] - *Architecting the next generation of service-based SCADA/DCS system of systems*. *IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society*. P. 359-364, 7-10 Novembro 2011, Karlsruhe, Alemanha, [citado em 21-05-2012].

Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=06119279>>.

ISBN 978-1-61284-972-0

[Lipnickas, 2009]

Lipnickas, Arunas [et al.] - *Interoperability of SCADA System Applications with Web Services*. *IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*. P. 196-200, 21-23 Setembro 2009, Rende (Cosenza), Itália, [citado em 18-05-2012].

Disponível

em:

<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5342997>>.

ISBN 978-1-4244-4881-4

[Phaithoonbuathong, 2010]

Phaithoonbuathong, Punnuluk; Harrison Robert; West Andrew; Monfared Radmehr, Kirkham Thomas - *Web Services-based automation for the control and monitoring of production systems*. *Internacional Journal of Computer Integrated Manufacturing [online]*. Vol. 23, No. 2, P. 126-145, 22 de Janeiro de 2010 [citado em 18-09-2011].

Disponível em: <<http://dx.doi.org/10.1080/09511920903440313>>.

[Quintã, 2008]

Quintã, André Figueiredo – *Integração de sistemas de produção*. Aveiro, Universidade de Aveiro, 2008. Tese de dissertação apresentada à Universidade de Aveiro para a obtenção do grau de Mestre em Engenharia Mecânica – Área de Especialização em Automação e Robótica.

[Reckziegel, 2006]

Reckziegel, Mauricio. – *Descrevendo, descobrindo e integrando Serviços web – UDDI*. 02 Agosto 2006. [citado em 28-09-2011]. Disponível em: <http://imasters.com.br/artigo/4474/webservices/descrevendo_descobrimdo_e_integrando_web_services_uddi/>.

[SCADA - TechnologyUK]

Supervisory Control and Data Acquisition (SCADA). Technology UK. [citado em 12-10-2011]

Disponível em:

<http://www.technologyuk.net/telecommunications/industrial_networks/scada.shtml>

[Shklar, 2003]

Shklar, Leon; Rosen, Richard – *Web Application Architecture Principals, protocols and practices*. John Wiley & Sons Ltd, 2003. 357 páginas.

ISBN 0-471-48656-6.

[SOAP 1.1]

Box, Don; Ehnebuske, David; Kakiyaya, Gopal; Layman, Andrew; Mendelsohn, Noah; Nielsen, Henrik Frystyk; Thatte, Satish; Winer, Dave – *Simple Object Access Protocol (SOAP) 1.1*. 8 Maio 2000. [citado em 28-09-2011].

Disponível em: <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/#_Toc478383486>.

[Tech-FAQ]

Tech-FAQ – SCADA. [citado em 12-10-2011].

Disponível em: <<http://www.tech-faq.com/scada.html>>.

[Tidwell, 2001]

Tidwell, Doug; Snell, James; Kulchenko, Pavel – *Programming Web Services with SOAP*. 1.º Edição. O’Reilly, Dezembro 2001. 216 páginas.
ISBN 0-596-00095-2.

[UDDI-W3C]

Sabbouth, Marwan [et al.] - *Workshop on Web services*. World Wide Web Consortium. 11-12 Abril 2001, San Jose, Califórnia, EUA, [citado em 28-09-2011].

Disponível em:

<<http://www.w3.org/2001/03/WSWS-popa/paper08>>.

[Wang, 2007]

Wang, Shengwuei [et al.] - *A middleware for web service-enabled integration and interoperation of intelligent building systems*. Automation in Constrution. P. 112-121, 2007, [citado em 17-05-2012].

Disponível

em:

<<http://www.sciencedirect.com/science/article/pii/S0926580506000124>>.

[Web Service, 2004]

Booth, David [et al.] – *Web Services Architecture*. 11 Fevereiro 2004. [citado em 16-08-2012].

Disponível em: < <http://www.w3.org/TR/ws-arch/#whatis>>.

[Wikipedia-UDDI]

Wikipedia. – *Universal Description Discovery and Integration*. 18 Agosto 2011. [citado em 28-09-2011]. Disponível em: <<http://en.wikipedia.org/wiki/UDDI>>.

[WSDL 1.1]

Christensen, Erik; Curbera, Francisco; Meredith, Greg; Weerawarana, Sanjiva. – *Web Services Description Language (WSDL) 1.1*. 15 Março 2001. [citado em 28-09-2011].

Disponível em: <<http://www.w3.org/TR/wsdl>>.

[WS-Tutorial, 2011]

Web Service to Web Service Communication. 20 Julho 2011. [citado em 28-09-2011].

Disponível em: <<http://www.wstutorial.com/web-service-to-web-service-communication/>>.