



Universidade Nova de Lisboa  
Faculdade de Ciências e Tecnologia

## Departamento de Engenharia Electrotécnica

Plataforma para testes de conformidade de sistemas  
baseados em módulos conceptuais STEP

Por

Sérgio Miguel da Silva Onofre

Dissertação apresentada na Faculdade de Ciências e Tecnologia da  
Universidade Nova de Lisboa para obtenção do grau de Mestrado em  
Engenharia Electrotécnica, Sistemas e Computadores.

Orientador: **Doutor Ricardo Luís Rosa Jardim Gonçalves**

Lisboa

2007



## **Agradecimentos**

Este espaço é dedicado àqueles que deram a sua contribuição para que esta dissertação fosse realizada. A todos eles deixo aqui o meu agradecimento sincero.

Gostaria de começar por deixar aqui o meu agradecimento muito especial à minha namorada Rita Barata pelo apoio e compreensão nos momentos mais difíceis e pelo encorajamento que sempre me transmitiu durante a realização desta dissertação.

Gostaria também de agradecer aos meus pais e irmã pelo apoio que sempre têm demonstrado ao longo da minha vida, incentivando-me sempre a atingir novas metas.

Agradeço ao meu orientador Prof. Doutor Ricardo Gonçalves pela forma como guiou o meu trabalho e por todas as recomendações que foi dando ao longo do trabalho, permitindo que este trabalho contribuísse para o meu desenvolvimento pessoal. Quero agradecer sobretudo o facto de nesta fase final, mesmo não estando bem fisicamente, não ter deixado de me ajudar e contribuir para o bom sucesso deste trabalho.

Por último, deixo também aqui uma nota de apreço aos meus amigos e colegas de trabalho do GRIS, pelo bom ambiente de trabalho e partilha de conhecimento prestada e sobretudo pela amizade e entre ajuda que sempre me têm demonstrado.

## Sumário

O conjunto de normas STEP (ISO 10303) é reconhecido pela comunidade científica e industrial como sendo importante para a integração de sistemas em ambientes industriais. Com o crescimento da utilização da Internet para troca de informação e conhecimento, outras tecnologias têm vindo a ser cada vez mais utilizadas para os mesmos fins, sendo o XML, um dos casos mais notórios.

Atenta a este facto, a ISO desenvolveu uma norma (ISO 10303-Parte 28) para utilização de XML nos seus protocolos de aplicação. Contudo, esta norma não contempla toda a informação presente nos modelos STEP, devido à falta de expressividade da linguagem usada (XSD) para representar os modelos na tecnologia XML.

Esta dissertação propõe a extensão dos modelos XSD, obtidos utilizando a parte 28 como referência, com esquemas Schematron e código Java. Para verificar que os dados XML usados cumprem todas as especificações estabelecidas pelos modelos conceptuais STEP, e que os sistemas que trocam e usam estes dados são interoperáveis, é proposta uma metodologia para testes de conformidade e verificação de interoperabilidade, adaptando e estendendo a metodologia descrita nas partes 30s da norma STEP. Para a sua implementação, foi desenvolvida uma plataforma baseada em serviços para a Internet (SOA).

## **Abstract**

The scientific and industrial community recognizes STEP (ISO 10303) as an important family of standards for product data integration and exchange in industrial environments. However, with the increase of internet usage to exchange information and knowledge, other technologies have started to gain importance for these purposes, being the XML, one of the most used.

Aware of these advances, ISO developed a standard (ISO 10303-Part 28) to use XML with its STEP application protocols. However, the transformation proposed by this standard does not support all the information contained in STEP models, since XSD is less expressive modelling language. This way it does not include all the information available in the original STEP models.

This dissertation proposes an extension to the XSD models, obtained using part 28 as reference, with Schematron schemas and Java code. To verify that the systems that implement STEP XML data are in conformance with all the specifications established by STEP conceptual models, and they are interoperable, this dissertation also proposes a methodology to conformance testing and interoperability checking, extending and adapting the methodology described in part 30 of STEP. It was developed a platform based in web services (SOA), to put this methodology in practice.

## Simbologia e Notações

AP	Application Protocol
ATC	Abstract Test Case
ATS	Abstract Test Suite
CE	Comissão Europeia
CT	Conformance Testing
ETSI	European Telecommunications Standards Institute
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IC	Interoperability Checking
ISO	International Organization for Standardization
IUT	Implementation Under Test
OSI	Open Systems Interconnection
PICS	Protocol Implementation Conformance Statement
PIXIT	Protocol Implementation extra Information for Testing
PME	Pequenas e médias empresas
SOA	Service Oriented Architecture
SUT	System Under Test
STEP	Standard for the Exchange of Product Model Data
TIC	Tecnologias de Informação e Comunicação
TTCN	Tree and Tabular Combined Notation
TTCN-3	Testing and Test Control Notation
WSDL	Web Services Description Language

XML	eXtensible Markup Language
XSD	XML Schema Definition
XSL	Extendible Stylesheet Language
XSLT	XSL Transformations

# Índice de Matérias

ÍNDICE DE FIGURAS.....	9
ÍNDICE DE QUADROS.....	10
INTRODUÇÃO.....	11
<b>1. MOTIVAÇÃO E CONTEXTO.....</b>	<b>17</b>
1.1. Normas de referência.....	19
1.1.1. A Importância da norma ISO 10303 – STEP.....	21
1.2. Testes de conformidade e verificação de interoperabilidade.....	22
1.3. Enquadramento da dissertação em projectos europeus de investigação.....	24
<b>2. MODELOS CONCEPTUAIS STEP E O XML.....</b>	<b>26</b>
2.1. Descrição da tecnologia STEP.....	26
2.1.1. Integração STEP-XML.....	29
2.2. Tecnologias para representação de regras e restrições em XML.....	31
2.2.1. Schematron.....	32
2.2.2. Constraint Markup Language (XCML).....	33
2.2.3. XML Constraint Specification Language (XCSL).....	34
2.2.4. Análise das linguagens descritas.....	34
2.3. Transformação de EXPRESS para Schematron.....	37
2.3.1. Mapeamento dos modelos STEP.....	38
2.4. Arquitectura da ferramenta para mapeamento de STEP-XML.....	41
<b>3. METODOLOGIAS E PLATAFORMAS PARA TESTES DE CONFORMIDADE E VERIFICAÇÃO DE INTEROPERABILIDADE.....</b>	<b>43</b>
3.1. Metodologias e Plataformas existentes.....	43
3.1.1. ISO 9646 - “OSI Conformance Testing Methodology and Framework”.....	43
3.1.2. INCITS 1703-D, Conformance Testing Methodology for ANSI/INCITS 358-2002, BioAPI Specification.....	45
3.1.3. ISO 10303 parts 30 – <i>Conformance testing methodology and framework</i> .....	47
3.1.4. ETS 300 406 - Methods for Testing and Specification (MTS).....	48
3.1.5. TS 102 237-1; Interoperability test methods & approaches.....	49
3.2. Análise comparativa das metodologias.....	51
<b>4. TESTES DE CONFORMIDADE E VERIFICAÇÃO DE INTEROPERABILIDADE PARA PROTOCOLOS DE APLICAÇÃO STEP REPRESENTADOS EM XML.....</b>	<b>54</b>
4.1. Metodologia para testes de conformidade.....	54
4.2. Plataforma SOA para testes de conformidade.....	57
4.3. Metodologia e plataforma de verificação de interoperabilidade.....	61
4.4. Integração dos testes de conformidade e verificação de interoperabilidade.....	63
<b>5. VALIDAÇÃO E DEMONSTRAÇÃO EM AMBIENTES INDUSTRIAIS.....</b>	<b>66</b>
5.1. Cenário de aplicação, e-Procurement.....	67
5.2. Cenário de aplicação, distribuição e entrega de produtos.....	71
<b>CONCLUSÕES E PERSPECTIVAS DE TRABALHO FUTURO.....</b>	<b>78</b>
<b>BIBLIOGRAFIA.....</b>	<b>82</b>
<b>ANEXO 1 – TABELAS DE MAPEAMENTO.....</b>	<b>86</b>
<b>ANEXO 2 – EXEMPLO DE TRANSFORMAÇÃO DE UM MODELO STEP.....</b>	<b>88</b>

## Índice de Figuras

Figura 1.1 – Progresso e evolução de serviços <i>online</i> .	12
Figura 1.1 – Globalização, criação de um mercado mundial.	17
Figura 1.2 – Conformidade e interoperabilidade entre sistemas.	23
Figura 1.3 – Enquadramento temporal do Mestrado.	24
Figura 2.1 – Exemplo de EXPRESS.	27
Figura 2.2 – Exemplo de instanciação em Parte 21	29
Figura 2.3 – Exemplo de XSD	30
Figura 2.4 – Transformação do modelo EXPRESS	31
Figura 2.5 – Exemplo de Schematron	32
Figura 2.6 – Exemplo de XCML	33
Figura 2.7 – Exemplo de XCSL	34
Figura 2.8 – Exemplo da regra UNIQUE e WHERE em EXPRESS	38
Figura 2.9 – Exemplo da regra UNIQUE em Schematron	39
Figura 2.10 – Exemplo da regra WHERE em Schematron	39
Figura 2.11 – Exemplo de uma função em EXPRESS	40
Figura 2.12 – Função transformada para Java	41
Figura 2.13 – Arquitectura proposta para transformação de modelos STEP	42
Figura 3.1 – Diagrama adaptado das fases da metodologia para CT da norma ISO 9646	44
Figura 3.2 – Diagrama adaptado das fases do processo de testes de conformidade para BioAPI	46
Figura 3.3 – Diagrama adaptado da metodologia usada para testes de conformidade de dados STEP	47
Figura 3.4 – Diagrama adaptado da metodologia de testes de conformidade da ETSI	49
Figura 3.5 – Diagrama adaptado da metodologia de verificação de interoperabilidade da ETSI	50
Figura 4.1 – Diagrama da metodologia proposta de testes de conformidade	55
Figura 4.2 – Serviços de testes de conformidade	58
Figura 4.3 – Exemplo de transformação	59
Figura 4.4 – Arquitectura proposta da plataforma de testes de Conformidade	60
Figura 4.5 – Exemplo de WSDL gerado	61
Figura 4.6 – Aplicação da metodologia para verificação de interoperabilidade	63
Figura 4.7 – Arquitectura conjunta para testes de conformidade e verificação de interoperabilidade	64
Figura 5.1 – Cenário 1 – e-Procurement	67
Figura 5.2 – Exemplo de duas entidades do pedido de orçamento	68
Figura 5.3 – XSD das entidades <i>Products_information</i> e <i>Product</i>	69
Figura 5.4 – Regras descritas em Schematron	70
Figura 5.5 – Exemplo erro detectado pelo CT	70
Figura 5.6 – Cenário 2 – Distribuição e entrega de produtos	72
Figura 5.7 – Exemplo da elaboração do modelo STEP através da descrição da FEDEX	75
Figura 5.8 – XSD e Schematron obtidos a partir do exemplo	76
Figura 5.9 – Exemplo de troca de dados entre as empresas	77

## Índice de Quadros

Tabela 2.1 – Quadro de comparações entre as linguagens.....	37
Tabela 3.1 – Quadro de comparações entre as metodologias.....	52

## Introdução

A evolução tecnológica e liberalização do comércio além fronteiras, tem levado a que o desenvolvimento de negócios esteja a passar de puramente local, regional ou nacional, para internacional ou mesmo global. Assim, sobretudo as pequenas e médias empresas (PME), devido à sua dimensão têm vindo a sentir necessidade de cooperar entre si para aumentar a sua competitividade e conseguir desta forma superar estes novos desafios. Para esta cooperação entre empresas estão tipificadas algumas dificuldades, sendo uma das principais, a falta de interoperabilidade entre os seus sistemas informáticos [1].

O IEEE [2] define a interoperabilidade como a “capacidade que dois ou mais sistemas têm de trocar informação e conseguir usar a informação que foi trocada”. O problema da interoperabilidade nas empresas pode ter diversas origens, como por exemplo, estas usarem diferentes tipos de serviços, diferentes sistemas para gestão de dados, diferentes modelos conceptuais para descrição de dados ou diferentes formatos para representação de dados [3]. Alcançar a interoperabilidade entre sistemas não é só uma preocupação das empresas que pretendem cooperar, mas também tem vindo a ser uma das principais preocupações dos governos e da Comissão Europeia como parte de uma estratégia económica global [4].

Seguindo de perto a globalização dos mercados e evoluções tecnológicas, a Comissão Europeia (CE) tem vindo a estabelecer como prioridade, o uso de novas tecnologias para modernizar a sua economia. Para tal, a CE tem vindo a definir estratégias e medidas, tal como a estratégia de Lisboa, para potenciar o desenvolvimento nas áreas tecnológicas, científicas, económicas e sociais. A CE pretende estimular a melhoria dos serviços públicos e comércio electrónico, a criação de uma rede transeuropeia de alta velocidade para as comunicações científicas via electrónica, interligando universidades, institutos de investigação, bibliotecas científicas, e a democratização do acesso à Internet e produção de conteúdos que valorizem o património cultural e científico [5].

Como exemplo desta continua preocupação, recentemente foi lançado o estudo “*Online Availability of Public Services: How Is Europe Progressing?*” [6], pela Comissão Europeia, que relata a situação da Europa em termos de novas tecnologias, mais concretamente a disponibilização de serviços públicos *online*. Este estudo define uma escala de disponibilização de serviços básicos *online* com cinco níveis:

1. Ausência de serviços;
2. Informação *online*;
3. Interação unilateral;
4. Interação bilateral;
5. Troca completa de dados.

Os países estudados<sup>1</sup> foram enquadrados dentro destes níveis, tendo em conta dados recolhidos entre 2001 e 2006, sendo possível observar na figura seguinte que a disponibilização destes serviços cresceu nos últimos seis anos de 45% para 78%. Pode-se assim concluir que a evolução tecnológica que a estratégia de Lisboa propõe tem vindo pouco a pouco a ter resultados.

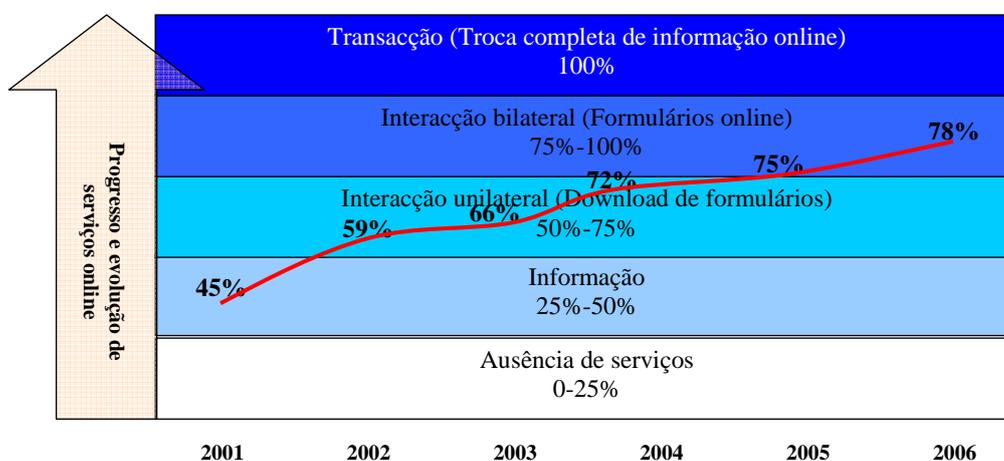


Figura 1.1 – Progresso e evolução de serviços *online*.

No entanto, não basta fomentar o uso deste tipo de serviços e potenciar a cooperação entre empresas. Deve também ser assegurada que esta crescente evolução seja acompanhada de

<sup>1</sup> Estados membros iniciais da União Europeia mais a Islândia, Suíça e Noruega

meios que permitam garantir a fiabilidade e confiança por parte de utilizadores e empresas, em serviços, produtos ou na troca de informação e conhecimento entre sistemas.

A adopção de uma norma de referência e posterior verificação da conformidade dos serviços ou dados pode dar uma grande ajuda no alcance da fiabilidade e confiança procurada. A conformidade com uma norma consiste em avaliar se um produto, processo ou serviço cumpre com os requisitos de referência definidos por esta. Assim, a normalização tem um papel importante, quer no alcance da interoperabilidade, visto que a adopção de normas por parte das empresas permite a estas partirem de uma referência comum já acordada a nível internacional, quer no aumento da confiança e fiabilidade nos serviços usados e dados de produto trocados, dado que a adopção de normas permite uma certificação através de organismos idóneos, tal como a ISO (*International Organization for Standardization*) [7][8].

De entre as várias normas existentes para representação de dados industriais, o conjunto de normas STEP (ISO 10303) é reconhecido pela comunidade internacional como uma referência para a integração e troca de dados relativos ao produto [9]. A norma STEP tem sido desenvolvida com o intuito de reduzir os problemas de interoperabilidade em sistemas que pretendam trocar dados de produto. O seu uso para reduzir estes problemas em diferentes áreas aplicacionais, diferenciadas em Protocolos de Aplicação (AP) tais como a indústria automóvel (AP214), indústria do mobiliário (AP236) ou construção (AP225), faz com que esta esteja a ser adoptada por diversas empresas [10].

Sendo usada como norma de referência para troca de dados por grandes empresas industriais, como a Boeing, a Renault ou General Motors, a norma STEP está bastante difundida no mercado. No entanto, a ISO 10303 usa geralmente a sua Parte 21 [11] para representação de dados, e como esta não se encontra num formato muito difundido nem de fácil implementação, tem vindo a dificultar uma maior implementação por parte das empresas ao STEP.

Em contrapartida, devido à cada vez maior utilização da Internet para troca de informação e conhecimento, têm vindo a surgir novas normas para representação de dados [12]. Estas normas, têm vindo a ganhar cada vez mais adeptos, sendo um dos casos mais evidenciados o XML (*eXtended Markup Language*) [13].

O XML deve a sua grande utilização à sua simplicidade e flexibilidade de implementação, facilidade de interpretação e validação da estrutura dos seus documentos através de um esquema de referência definido para o efeito, que resultou no número elevado de aplicações informáticas que têm adoptado XML nas aplicações disponíveis no mercado [14].

Para acompanhar esta tendência crescente na adopção do XML por parte das empresas, a ISO, desenvolveu normas que descrevem regras para transformação de modelos STEP para tecnologias de implementação emergentes, como é o caso da Parte 28 (ISO 10303-Parte 28) que descreve a transformação de EXPRESS (ISO 10303-Parte 11) [15], linguagem que é usada na descrição dos modelos STEP, para XSD (*XML Schema Definition*) [16], uma das linguagens que descreve os modelos na tecnologia XML. Esta norma permite que as empresas adoptem os protocolos de aplicação de normas STEP ao mesmo tempo que usam o XML para troca de dados de produto.

No entanto, a parte 28 do STEP, por condicionantes do próprio XSD, não permite abranger a representação de toda a informação que um protocolo de aplicação STEP pode conter. Por exemplo, as regras “*WHERE*” não estão consideradas, visto o XSD, não suportar a definição deste tipo de regras. Com vista a desfazer esta condicionante, quando se transformam os modelos descritos em EXPRESS em XSD, no capítulo dois desta dissertação é proposta uma arquitectura que permite definir regras para a transformação dos modelos STEP para uma tecnologia que consiga complementar o XSD, e assim suportar toda a informação contida em modelos descritos em EXPRESS.

De entre várias linguagens existentes para suportar regras sobre XML, foram analisadas algumas, sendo que o Schematron [17] foi a escolhida para descrever as regras e restrições

presentes nos modelos STEP e completar assim o XSD. Conseguindo transformar os modelos STEP em modelos equivalentes na tecnologia XML, executando testes de conformidade, é possível validar os dados em XML contra os modelos descritos pela norma STEP.

A tecnologia STEP descreve textualmente nas partes 30s uma metodologia para execução de testes de conformidade, com vista a garantir que os dados de produto estejam em conformidade com a norma. Esta metodologia não é a única que existe para testes de conformidade e no terceiro capítulo desta dissertação é apresentado um estudo de algumas metodologias para realização destes testes, sendo feita no final desse capítulo uma análise comparativa das mesmas.

Em resultado do estudo apresentado no capítulo três, no capítulo quatro é proposta uma metodologia para testes de conformidade de informação documentada em suporte XML com base em modelos definidos em STEP, sendo esta uma sistematização da descrita nas partes 30 da ISO, completada com uma nova extensão que integra características relevantes encontradas nas outras metodologias estudadas. De seguida é proposta uma metodologia para verificação de interoperabilidade entre dois sistemas que adoptem normas STEP e usem o XML como formato para troca de dados.

Este capítulo termina descrevendo uma plataforma para testes de conformidade e verificação de interoperabilidade baseada em SOA (Arquitetura Orientada a Serviços). A adopção desta plataforma SOA permite o fácil acesso aos serviços disponibilizados para testes de conformidade e verificação de interoperabilidade por parte das empresas independentemente da sua localização, possibilitando também incorporar facilmente estes serviços nas suas aplicações. A incorporação destes serviços estimula a um maior aumento da adesão das empresas à adopção de normas ISO, facilitando assim a certificação dos seus produtos ou serviços.

No último capítulo da dissertação, são descritas duas aplicações práticas que utilizam as metodologias propostas e a plataforma desenvolvida, onde através de cenários de demonstração se validam as contribuições desta dissertação.

Por último, são apresentadas conclusões sobre o trabalho de dissertação desenvolvido e discutidos os resultados obtidos da aplicação deste trabalho nos cenários descritos. A dissertação termina tecendo algumas considerações relativas à expectativa de trabalho futuro.

## 1. Motivação e contexto

Nos últimos anos tem vindo a notar-se cada vez mais o fenómeno chamado Globalização. Este pode definir-se como o processo que visa unir o mundo em todos os seus aspectos, sendo caracterizado pela uniformização dos centros urbanos, pelo alargamento das empresas ou organizações além fronteiras e pela evolução tecnológica, permitindo o entrosamento entre países, povos e culturas [18].

A nível comercial, a Globalização desafia uma maior integração e liberalização do comércio além fronteiras, fazendo com que as barreiras para interligação do comércio internacional venham a diminuir, criando assim um mercado único mundial. Este, com interdependência de mercados e produtores de diferentes países, poderá permitir maiores ganhos para os mercados internos já saturados [19].

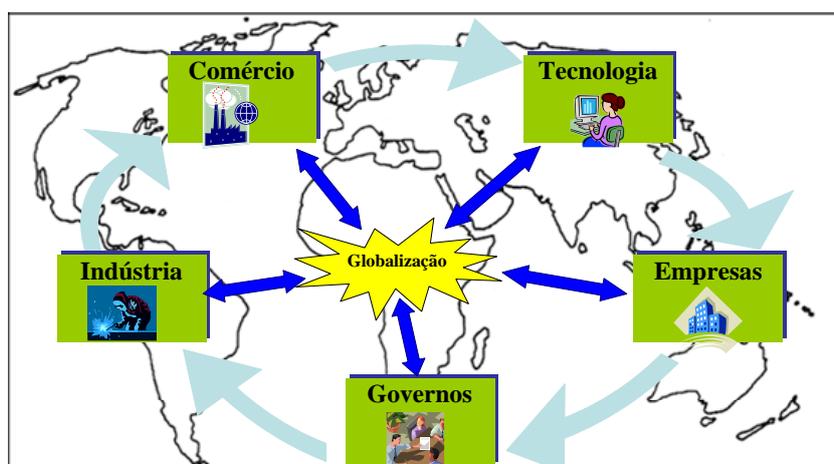


Figura 1.1 – Globalização, criação de um mercado mundial.

Para a concretização deste mercado mundial, vários sectores devem ser considerados, como:

- As indústrias, necessitando de se adaptar a este novo mercado, fazendo com que os seus produtos preencham os requisitos que novos clientes possam trazer;
- As empresas, estabelecendo novas parcerias e formas de interagir entre elas para competir com outras empresas do seu sector;

- Os governos, criando iniciativas para que os seus países possam acompanhar este desenvolvimento;
- O comércio, adaptando-se a novos processos de venda, como a venda online;
- A tecnologia, disponibilizando meios para viabilizar todos estes progressos.

A Comissão Europeia (CE) na estratégia de Lisboa, tendo como uma das suas preocupações que a Europa não se atrase neste processo de Globalização, define um conjunto de medidas em diferentes áreas de acção para tornar a Europa, no ano 2010, numa economia mais competitiva a nível mundial [5].

Mais recentemente, a CE lançou mais uma iniciativa com vista a dar continuidade à estratégia de Lisboa, o i2010 – *European Information Society for growth and employment* [20], que sucede à acção eEurope 2005 [21], mostrando que continua a ser uma prioridade para a CE impulsionar uma economia mais aberta e informatizada, realçando o uso generalizado das Tecnologias de Informação e Comunicação (TIC) nos serviços públicos, nas PME (Pequenas e Médias Empresas) e a nível doméstico.

No que diz respeito à integração de sistemas TIC por parte das empresas, ainda existem algumas dificuldades, visto ainda existir uma grande falta de interoperabilidade entre eles. Uma das formas de se contribuir para a interoperabilidade entre sistemas é a adopção de normas, assegurando assim, através do uso destas, uma uniformidade entre os dados e sistemas [22].

No advento da globalização, alcançar a interoperabilidade entre sistemas e aplicações e conformidade com as normas, é uma das grandes preocupações das empresas no uso de sistemas TIC. Estas procuram que a troca de dados de produto com outras empresas seja feita sem problemas, isto é, garantindo que não existam erros nos dados trocados, que haja uma compreensão completa da informação trocada, e que estes cumpram com as imposições do modelo estabelecido.

Para assegurar uma correcta troca de dados de produtos entre as empresas, devem ser implementados dois tipos de acções: Testes de Conformidade e Verificação de Interoperabilidade. Os testes de conformidade asseguram a correcta instanciação de dados de produtos em relação a uma norma. A verificação de interoperabilidade garante que os sistemas interligados consigam trocar os dados com semântica correcta. Deste modo, para que as organizações consigam acompanhar o crescente ritmo imposto pela globalização, utilizando cada vez mais sistemas TIC e sem perder confiança na troca de dados de produto, a implementação de testes de conformidade e verificação de interoperabilidade pode dar o suporte necessário a consegui-lo.

Para as empresas que usam normas STEP e pretendam adoptar o XML como formato nas suas trocas de dados ou mesmo para aquelas empresas que já usam XML, mas querem certificar os seus dados com normas STEP, esta dissertação pretende contribuir para que estas consigam assegurar que os dados de produtos estejam em conformidade com as normas STEP e os sistemas que usam ou manipulam estes dados sejam interoperáveis.

Assim, a motivação desta dissertação é ajudar as empresas na utilização e implementação de sistemas que usem conjuntamente dados XML e modelos STEP, propondo uma plataforma para testes de conformidade e verificação de interoperabilidade.

### **1.1. Normas de referência**

Uma norma, segundo a ISO (*International Organization for Standardization*), “é um documento aprovado por um organismo oficialmente reconhecido para tal, que, através do uso comum e repetitivo, estabelece regras, directivas ou características de produtos, processos ou serviços” [23]. As normas contêm especificações técnicas ou critérios específicos que ajudam a regular, através de uma maneira sistemática e lógica [23].

A utilização de normas de referência na representação de dados pode ajudar a conseguir a interoperabilidade entre sistemas, mas os benefícios do uso de normas vai para além desse

aspecto. Por exemplo, permitir reduzir esforços de integração, visto ser mais fácil integrar dois sistemas que usem a mesma norma como base, ou reduzir custos em implementações, visto existir um maior número de soluções genéricas disponíveis no mercado que podem ser usadas ou adaptadas a um caso particular.

A importância de uma norma, pode revelar-se maior quando esta é uma norma internacional, visto poder ter uma abrangência muito maior e diversificada. As normas internacionais podem aumentar a confiança e efectividade em bens e serviços que são usados no dia a dia. Podem oferecer uma ajuda na integração de meios no mundo do trabalho, regulando operações ou transferências de dados entre empresas, contribuindo, por exemplo, para reduzir custos, reduzir tempo de manufactura, aumentar a qualidade e eliminar dados redundantes [23].

Representando 157 países, a ISO, é o maior organismo no desenvolvimento de normas, tendo publicado mais de 16 mil normas internacionais. Estas normas, têm repercussões a nível social e económico, beneficiando não só os engenheiros e produtores, mas também a sociedade em geral, visto conseguir aumentar a confiança nos produtos e serviços que são usados.

Em processos de negócios, a adopção de normas internacionais ISO faz com que os fornecedores possam desenvolver os seus produtos e serviços com especificações adequadas ao seu sector de mercado, abrindo também a porta para mercados internacionais. Por outro lado, ao existir uma maior variedade de oferta de produtos e serviços e uma maior concorrência para oferecer os melhores produtos ao mais baixo preço, os clientes também beneficiam deste facto [23].

Para além de fazer com que os clientes ou utilizadores tenham uma maior confiança nos produtos e serviços certificados por uma norma internacional, permite a liberalização do mercado mundial, a interligação de bens entre os vários sectores industriais, a

implementação de sistemas de comunicação mundial e satisfazer a necessidade de normas globais para as tecnologias emergentes [23].

As normas ISO cobrem várias áreas, como a agricultura, construção, engenharia mecânica, dispositivos médicos, tecnologias de informação, etc.. Estas estão presentes em várias destas áreas indústrias, onde é possível encontrar a norma ISO 10303, mais conhecida como STEP (*STandard for the Exchange of Product model data*), sendo utilizada com o objectivo de normalizar a troca de informação de produtos.

### **1.1.1. A Importância da norma ISO 10303 – STEP**

A norma ISO 10303 – STEP, cujas características técnicas são descritas em maior detalhe no capítulo seguinte, tem vindo a ganhar importância em vários sectores da indústria, visto conseguir suportar informação do produto no seu ciclo de vida, quer estes dados sejam, por exemplo, da indústria de mobiliário, de electrónica ou de construção [24].

Esta é uma forte alternativa para substituir as várias propostas proprietárias que existem para representar dados de produto. O uso desta norma traz benefícios como a reutilização de dados, partilha e troca de dados, disponibilidade dos dados em todo o ciclo de vida dos produtos e processos, integração de dados, gestão partilhada de configurações e vários níveis de detalhes de informação de produtos [25].

STEP é de grande importância em vários sectores do mercado industrial, como na indústria aeroespacial, na indústria automóvel, indústria do petróleo e gás, ou na indústria do mobiliário. A grande importância desta norma, na indústria, deve-se a benefícios de uso e implementação como [23]:

- Troca de dados neutros entre sistemas heterogéneos;
- Uso de dados neutros para trocas entre clientes e fornecedores;
- Eliminação da necessidade de traduções directas;
- Aumento da qualidade dos produtos e redução do tempo de desenvolvimento;

- Integração empresarial através de bases de produto;
- Negócios entre várias organizações;
- Suporte para manutenção do ciclo de vida do produto.

Também no caso específico das PME, que por vezes sentem mais dificuldades em cooperar entre si ou com as grandes empresas, devido à sua dimensão e limitação de recursos, as normas STEP podem ter um papel facilitador na integração destas organizações em cenários colaborativos. Assim, quando as PME fazem parte de grandes cadeias de produção, produzindo componentes para as grandes empresas, a adopção da norma STEP pode facilitar a troca de dados de produtos (como por exemplo, design ou especificações do produto), definido um formato normalizado para a troca.

## **1.2. Testes de conformidade e verificação de interoperabilidade**

A aplicação de uma norma por si só não significa que todos os problemas de interoperabilidade deixem de existir, nem que os seus dados estejam em conformidade. Isto significa que se por exemplo duas empresas adoptarem a mesma norma, não implica que estas consigam trocar dados entre si, e que esses dados não contenham erros, quer a nível sintáctico quer a nível semântico. Assim, a aplicação de normas deve ser acompanhada por testes de conformidade e por verificações de interoperabilidade [26].

Segundo a “*ISO/IEC Guide 25*” a conformidade de um produto, processo ou serviço com a norma que o rege, é quando este preenche todas as especificações relevantes dos seus requisitos de conformidade [24]. Deste modo, os testes de conformidade (CT<sup>2</sup>), são testes que são feitos a um determinado produto, processo ou serviço com vista a concluir se este está conforme ou não com a especificação técnica que o rege.

---

<sup>2</sup> CT do termo em inglês *Conformance testing*

“Interoperabilidade é a capacidade que dois ou mais sistemas ou componentes (informatizados ou não) têm de trocar informação e usar essa mesma informação” [2]. A verificação de interoperabilidade (IC<sup>3</sup>) é feita de forma a garantir que um sistema consiga interoperar com outros.



Figura 1.2 – Conformidade e interoperabilidade entre sistemas.

Os testes de conformidade e a verificação de interoperabilidade são procedimentos que devem ser efectuados para verificar e assegurar a qualidade de um sistema, quer como procedimentos de monitorização, quer quando uma nova aplicação é adicionada ao mesmo. O CT é normalmente aplicado a um sistema isolado, enquanto o IC é aplicado entre sistemas.

Os benefícios de usar testes de conformidade na troca de dados de produto entre empresas ou na certificação de um produto, podem ser vistos de várias perspectivas, e.g., dos vendedores ou fabricantes e dos compradores ou utilizadores [27].

<sup>3</sup> IC do termo em inglês *Interoperability Checking*

### 1.3. Enquadramento da dissertação em projectos europeus de investigação

O trabalho desenvolvido no âmbito deste mestrado esteve directamente relacionado com o organismo de normalização internacional ISO, e foi implementado e validado em projectos de investigação e desenvolvimento europeus. Na figura seguinte é possível ver representado o diagrama temporal que relaciona o trabalho desenvolvido neste mestrado com o trabalho desenvolvido nestes projectos e para a norma internacional ISO 10303 AP236.



Figura 1.3 – Enquadramento temporal do Mestrado.

O trabalho desenvolvido nesta dissertação começou a ser desenvolvido no âmbito do projecto SMART-fm “A standards-compliant framework to support complete integrated product life-cycle information management and electronic commerce for the furniture manufacturing (fm) industry, in the advent of smart enterprise” cujo objectivo geral era identificar os requisitos necessários para preparar a indústria e a comunidade científica para o processo de globalização do comércio; estimular a indústria do mobiliário para novos modelos de negócio e organização, bem como para a adopção de normas, novas tecnologias e práticas de trabalho [28].

Este projecto fazia parte da iniciativa funSTEP que tem vindo a trabalhar no sentido de resolver os problemas de interoperabilidade entre as diversas empresas do panorama europeu do sector mobiliário [29]. Esta iniciativa tem vindo também a trabalhar junto da ISO, tendo desenvolvido o protocolo de aplicação AP236 (ISO 10303-236 - STEP AP236 -

*furniture catalog and interior design*). O trabalho de dissertação foi utilizado na validação desta norma internacional [30].

O trabalho desenvolvido nesta dissertação também teve uma participação importante no projecto ATHENA “*Advanced Technologies for interoperability of Heterogeneous Enterprise Networks and their Applications*”, cujo objectivo era contribuir para ultrapassar barreiras na área de interoperabilidade, transferindo e aplicando os resultados da pesquisa no projecto para vários sectores industriais, como o automóvel, aeronáutico, mobiliário e telecomunicações. O projecto implementa também sistemas piloto e treino para validar a abordagem tomada, promovendo uma nova cultura de negócio em rede e contribuindo para a criação de uma estrutura abrangente de interoperabilidade. [31].

## 2. Modelos Conceptuais STEP e o XML

Hoje em dia, a norma STEP é bastante popular em ambientes indústrias para representação e integração de dados relativos ao produto. Por outro lado, o XML é cada vez mais o formato escolhido para troca de dados na Internet.

A possibilidade de usar estas duas tecnologias em paralelo, isto é, usar o STEP como modelo de referência, acedendo assim às várias normas presentes na ISO, e o XML como formato para trocar de dados entre sistemas informáticos, é discutida neste capítulo. É assim proposta uma ferramenta de software que permite estabelecer uma ponte entre estas duas tecnologias, permitindo às empresas usar o melhor dos dois mundos.

### 2.1. Descrição da tecnologia STEP

A norma STEP (*STandard for the Exchange of Product model data*) cujo nome formal é ISO 10303: *Industrial automation systems and integration – Product data representation and Exchange*, foi desenvolvida com o objectivo de facilitar os mecanismos para partilha e troca de dados de produto.

A norma STEP é constituída por várias partes, tendo cada uma destas partes diferentes objectivos e propósitos. Entre outras, é constituída pelo conjunto de normas para trocas de produto (Protocolos de Aplicação, Partes 200s), pelos métodos que as descrevem e formas de implementação (Partes 10s e 20s), pelos mecanismos para verificação de conformidade dos dados (Partes 30s) e pelos recursos integrados (Partes 40s e 50s) [32].

Os Protocolos de Aplicação (AP<sup>4</sup>), para além de definirem um escopo de aplicação, também reúnem informação sobre normas e recursos que estes usam, a informação a ser trocada, testes a serem aplicados e directivas para a sua implementação [33].

---

<sup>4</sup> AP do termo em inglês *Application Protocol*

Para descrever os modelos presentes nos AP é usada a linguagem EXPRESS (ISO 10303-11) [15], que combina entidades, atributos e relações com conceitos de modelação por objectos. Esta linguagem permite a representação de herança entre entidades e contempla uma linguagem descritiva de regras e restrições no modelo [34] .

```
SCHEMA stand;

TYPE cores = ENUMERATION OF
  (amarelo,
   verde,
   azul,
   branco,
   vermelho,
   castanho,
   preto);
END_TYPE;

ENTITY automovel;
  marca: STRING;
  modelo: STRING;
  cor: SET [1:?] OF cores;
  matricula: STRING;
  ano: INTEGER;
  lugares: INTEGER;
  preco: INTEGER;
  iva: INTEGER;
  UNIQUE
    ur1: matricula;
  WHERE
    wr1: { 1885 <= ano <= 2007 };
    wr2: iva = 21;
END_ENTITY;

ENTITY comercial
  SUBTYPE OF (automovel);
  WHERE
    wr1: SELF\ automovel.lugares = 2;
END_ENTITY;

ENTITY todoTerreno
  SUBTYPE OF (automovel);
  traccao: INTEGER;
END_ENTITY;

ENTITY carro
  SUBTYPE OF (automovel);
  cabrio: BOOLEAN;
END_ENTITY;

ENTITY carrinha
  SUBTYPE OF (automovel);
  caixaAberta: BOOLEAN;
END_ENTITY;

END_SCHEMA;
```

Figura 2.1 – Exemplo de EXPRESS

Na figura anterior é possível ver um exemplo de um modelo descrito em EXPRESS, onde é descrito um modelo para um stand de automóveis. O modelo começa por definir um tipo “cores” (TYPE cores), onde estão enumeradas todas as cores que um automóvel pode ter.

Seguidamente, encontra-se uma entidade, “automovel” (ENTITY automovel), onde são descritos os principais atributos de cada automóvel, como a marca, o modelo, a sua cor, a matrícula, o ano, número de lugares, preço e IVA aplicado. Como se pode constatar, a cor do automóvel é definida utilizando o tipo acima descrito. Nesta entidade, para além dos atributos, também existem três regras, uma regra UNIQUE e duas WHERE.

O objectivo da regra *UNIQUE* é garantir que não se venham a definir dois automóveis com matrículas iguais. A primeira regra *WHERE* tem o objectivo de garantir que o ano do carro seja um ano real, isto é, depois de 1885, ano em que foi inventado o primeiro automóvel e inferior a 2007, sendo este o ano actual. A segunda regra pretende verificar se o IVA aplicado ao preço do automóvel é de 21%.

As entidades que se seguem são subtipos da entidade “automovel”, isto é, estas entidades têm os mesmos atributos que a entidade “automovel”, mas podem conter novos atributos ou regras especiais. Neste modelo, estas entidades são usadas para diferenciar diferentes categorias de automóveis.

No exemplo, a entidade “carro” (ENTITY carro) acrescenta mais um atributo “cabrio”, sendo este BOOLEAN (verdadeiro ou falso), utilizado para saber se o automóvel é descapotável ou não. A entidade “carrinha” (ENTITY carrinha) acrescenta também um atributo “caixaAberta”, também BOOLEAN, para especificar se a carrinha é de caixa aberta ou não.

A entidade “todoTerreno” (ENTITY todoTerreno) acrescenta um atributo “traccao” onde se pode definir se o automóvel todo o terreno tem tracção a duas ou a quatro rodas. Por último está descrita a entidade “comercial” (ENTITY comercial), que contém uma regra WHERE, sendo que esta regra estipula que os automóveis comerciais apenas podem ter dois lugares.

O formato tradicional para troca de dados em STEP é definido na Parte 21 (ISO 10303-21), sendo que este é baseado em caracteres ASCII [11]. Na figura seguinte é possível ver um exemplo de instanciação em Parte 21, onde se pode observar que cada entidade (automovel), é representada por um número precedido de #, e os seus atributos são descritos dentro de parênteses. Este formato, como se pode verificar, não é de fácil leitura humana e para além disso oferece dificuldades quando se tenta transformar noutra formato.

```
#2 = automovel('PEUGEOT', '207', '01-29-EX', (amarelo), 1990, 2, 20000, 21);  
#10 = automovel('BMW', '320D', '01-BI-89', (azul), 2006, 5, 100000, 21);  
#20 = comercial(#2);  
#100 = carrinha('BMW', '320D', '02-BI-89', (azul), 2006, 5, 30000, 21, false);
```

Figura 2.2 – Exemplo de instanciação em Parte 21

Neste exemplo estão instanciadas quatro entidades, duas “automovel”, uma “carrinha” e uma “comercial”. Na primeira linha, na entidade “automovel” é descrito um Peugeot 207 amarelo de cinco lugares com a matrícula 01-29-EX de 1990 e cujo preço é de 20000€ mais 21% de IVA. A segunda linha descreve um BMW 320D azul de 2006 com a matrícula 01-BI-89 com o valor de 100000€ mais 21% de IVA. Estas duas entidades definem os automóveis de uma forma geral, não dando informação se é um carro, carrinha, comercial ou todo o terreno.

Nas duas linhas seguintes, estão definidas duas entidades “carrinha” e “comercial”, que como se pode ver utilizando mesmas características (uma por referência e outra redefinindo os atributos), mais as específicas para cada categoria (caso seja necessário), conseguem dar também a informação de que tipo de automóvel se trata, neste caso um comercial e uma carrinha respectivamente.

### 2.1.1. Integração STEP-XML

Devido à dificuldade em usar a Parte 21 por parte da indústria, pela sua complexidade, e pela pouca disponibilidade de ferramentas informáticas no mercado que usem este formato, o uso de formatos como o XML têm vindo a ser preferidos.

O XML foi desenvolvido a partir do SGML (ISO 8879) com o objectivo de ir ao encontro das novas necessidades de publicação de dados electrónicos. É um formato de texto simples e flexível, usado sobretudo para troca de dados em aplicações Web. Este formato pode ser usado em inúmeras plataformas, sendo interpretado por uma grande variedade de ferramentas e dispondo de uma grande variedade de ferramentas para a sua manipulação [35].

Tomando em conta factores como este, a ISO rapidamente desenvolveu outras partes, como a Parte 28 (ISO 10303-28)[36], com o objectivo de tornar possível a instanciação de dados em XML mantendo como referência os modelos conceptuais STEP.

```
<xs:simpleType name="cores">
  <xs:restriction base="xs:normalizedString">
    <xs:enumeration value="amarelo"/>
    <xs:enumeration value="verde"/>
    <xs:enumeration value="azul"/>
    <xs:enumeration value="branco"/>
    <xs:enumeration value="vermelho"/>
    <xs:enumeration value="castanho"/>
    <xs:enumeration value="preto"/>
  </xs:restriction>
</xs:simpleType>

<!--ENTITY automovel-->
<xs:element name="automovel" type="st:automovel" block="extension restriction"
substitutionGroup="exp:Entity" nillable="true"/>
<xs:complexType name="automovel">
  <xs:complexContent>
    <xs:extension base="exp:Entity">
      <xs:sequence>
        <xs:element name="marca" type="xs:normalizedString"/>
        <xs:element name="modelo" type="xs:normalizedString"/>
        <xs:element name="cor"/>
        <xs:element name="matricula" type="xs:normalizedString"/>
        <xs:element name="ano" type="xs:long"/>
        <xs:element name="lugares" type="xs:long"/>
        <xs:element name="preco" type="xs:long"/>
        <xs:element name="iva" type="xs:long"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figura 2.3 – Exemplo de XSD

A Parte 28 descreve como fazer a transformação dos modelos definidos em EXPRESS para os seus equivalentes em *XML Schema Definition (XSD)*, facilitando assim a troca de dados em formato XML. Na figura anterior é ilustrada uma parte (tipo “cores” e a entidade

“automovel”) do modelo XSD obtido do exemplo de EXPRESS, descrito anteriormente neste capítulo para um Stand de automóveis, utilizando a parte 28 como referência.

No entanto, esta parte apenas contempla a transformação da parte sintáctica dos modelos, isto é, não engloba partes semânticas como as regras, restrições, funções ou procedimentos. A razão pela qual esta parte não contempla a parte semântica, é que o XSD não permite definir nos seus modelos este tipo de informação.

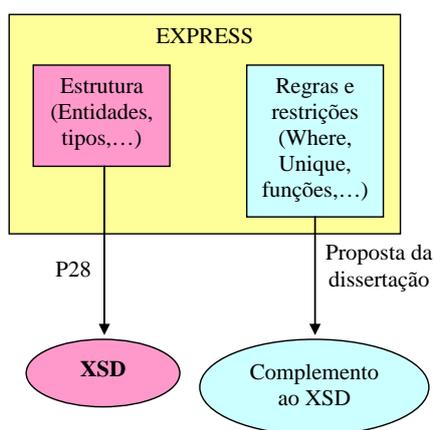


Figura 2.4 – Transformação do modelo EXPRESS

Assim, visto o XSD não suportar regras e restrições e não existir uma alternativa para suportar esta parte semântica dos modelos STEP, faz com que não seja possível uma validação completa dos dados XML contra os modelos conceptuais STEP. Torna-se assim necessário, tal como ilustrado na figura anterior, criar um complemento para o XSD para suportar a parte semântica dos modelos, tornando assim possível esta validação.

## 2.2. Tecnologias para representação de regras e restrições em XML

Como foi referido anteriormente, é possível transformar os modelos descritos em EXPRESS para XSD, usando como base as normas de transformação descritas na parte 28. Porém, esta parte deixa de fora as regras, restrições, funções e procedimentos que também é possível encontrar nestes modelos, sendo assim necessário existir uma alternativa que possibilite a análise completa dos dados em XML.

Para suportar este tipo de informação do modelo, regras e restrições sobretudo, e testar se estas estão a ser implementadas nos dados em XML, existem linguagens como o *Schematron* [17], *Constraint Markup Language (XCML)* [37] e *XML Constraint Specification Language (XCSL)* [38], entre outras, que permitem descrever este tipo de informação e ao mesmo tempo validá-la. Sendo estas três linguagens as mais populares, seguidamente é feita uma breve descrição de cada uma delas e uma posterior análise de qual a mais indicada a usar para representar a parte semântica dos modelos STEP com vista a validar os dados XML.

### 2.2.1. Schematron

O Schematron é uma linguagem XML, simples, estruturada e poderosa de validação. Esta linguagem faz inferências sobre a existência ou não de padrões em árvores. Sendo também uma linguagem esquemática, difere de outras no aspecto de não ser baseada em gramática, mas sim na procura de padrões em árvores de documentos. Descrita normalmente usando XPath<sup>5</sup>, permite a descrição de regras e restrições em forma de declarações. O Schematron tem vindo a evoluir e actualmente já é uma norma ISO/IEC (19757) [14].

```
<schema xmlns="regraStand">
  <title>Exemplo de Schematron</title>
  <pattern>
    <rule context="comercial">
      <assert test="lugares = 2">Erro, o comercial não tem dois lugares!</assert>
    </rule>
  </pattern>
</schema>
```

Figura 2.5 – Exemplo de Schematron

Na figura acima ilustrada, está definido um exemplo de um esquema em Schematron, onde está descrita uma regra cujo objectivo é procurar, em documentos XML, todas as ocorrências de entidades “comercial” e ver se em todas estas o atributo “lugares” contém o número dois. Se alguma das entidades não tiver o número dois no atributo “lugares” aparece a mensagem de erro “*Erro, o comercial não tem dois lugares!*”.

---

<sup>5</sup> <http://www.w3.org/TR/xpath>

Como se pode ver pelo exemplo a definição de regras nesta linguagem é bastante simples, bastando apenas conhecer a sintaxe do XPath para conseguir definir as regras. O Schematron pode ser usado em conjunto com outras tecnologias para verificação de estrutura como o XSD, TREX<sup>6</sup>, RELAX<sup>7</sup>, etc. No caso do XSD o Schematron pode ser incluído dentro do esquema como um elemento.

Para validar as regras descritas em Schematron normalmente são executados dois passos: O primeiro compila o documento contendo o esquema com as regras e restrições para um formato apropriado a executar a validação. O segundo passo consiste em validar o ficheiro XML contra este último gerado.

### 2.2.2. Constraint Markup Language (XCML)

Tal como o Schematron, a *Constraint Markup Language* (XCML), também é uma linguagem XML e a sua sintaxe é definida em XSD. Fácil de criar, simples e concisa, esta linguagem permite, não só definir regras estáticas, como também regras dinâmicas, do tipo *if-then-else*. Esta linguagem suporta parâmetros dinâmicos para definir regras dinâmicas. Para definir expressões esta linguagem também suporta o XPath.

```
<Constraint context="automovel">
  <Parameter>
    <name>taxa</name>
    <type>decimal</type>
    <defaultValue>21</defaultValue>
  </Parameter>
  <Assertion test="IVA=$taxa"/>
</Constraint>
```

Figura 2.6 – Exemplo de XCML

Na figura anterior, é ilustrado um exemplo de uma regra definida em XCML, onde o objectivo é testar se os atributos "IVA" das entidades "automovel" têm o valor 21%. A validação com

<sup>6</sup> <http://www.thaiopensource.com/trex/>

<sup>7</sup> <http://www.xml.gr.jp/relax/>

esta linguagem é feita da mesma forma que a validação do Schematron, isto é, seguindo os mesmos dois passos.

### 2.2.3. XML Constraint Specification Language (XCSL)

A linguagem *XML Constraint Specification Language* (XCSL) é semelhante às linguagens anteriormente descritas. Esta linguagem foi desenvolvida com o objectivo de escrever condições para restringir os valores presentes em XML. A XCSL permite definir declarações com as restrições que se pretende verificar nos documentos XML. Estas declarações são transformadas em XSLT<sup>8</sup>, sendo depois executadas sobre o documento. Assim, tal como as linguagens anteriormente estudadas, também precisa de dois passos para executar a validação.

Na figura seguinte é ilustrado um pequeno exemplo desta linguagem, onde se pretende verificar se o preço do automóvel está entre os zero euros e os cem euros.

```
<CS>
  <constraint>
    <selector selexp="/automovel/preco"/>
      <cc>(.>0) and (<.100)</cc>
      <action>
        <message>Preco incorrecto!</message>
      </action>
    </constraint>
  </CS>
```

Figura 2.7 – Exemplo de XCSL

### 2.2.4. Análise das linguagens descritas

Antes de escolher a linguagem a utilizar para representar as regras e restrições presentes nos modelos descritos em EXPRESS, foram analisadas as linguagens descritas anteriormente de acordo com os seguintes parâmetros: valor, escala, forma, acção, ponto de teste, facilidade de elaboração de regras, organização responsável, disponibilização de

---

<sup>8</sup> <http://www.w3.org/TR/xslt>

ferramentas para uso e integração com o XSD. Estes parâmetros são divididos em duas partes, a parte que avalia a capacidade das linguagens na construção de restrições [39] e a parte que avalia outras características gerais relevantes.

### **Construção de restrições:**

- Valor – restrições para valores de dados XML
  - Estático – Impõe as mesmas limitações para uma instância XML
  - Dinâmico – Impõe diferentes limitações para uma instância XML
- Escala – número de nós num documento XML que estão ou podem ser abrangidos pela restrição
  - Dados – considera um único atributo ou elemento simples
  - Tuplo – considera um conjunto de atributos ou elementos simples de um elemento complexo
  - Elementos – considera um conjunto de atributos ou sub-elementos de um elemento complexo
  - Repositório – considera um conjunto de atributos ou elementos de diferentes elementos complexos
- Forma – notação usada para descrever as restrições
  - Expressões booleanas – analisa um predicado e devolve verdadeiro ou falso
    - Simples – só um elemento ou atributo
    - Composta – dois ou mais elementos ou atributos
  - Regras condicionais – contêm expressões booleanas e podem alterar ou não os dados
    - Simples – só uma expressão condicional
    - Composta – duas ou mais grupos de expressões condicionais
- Acção – tipo de acção que pode efectuar quando uma restrição não é verificada
  - Informativa – mostra uma mensagem de erro
  - Activa – executa operações para corrigir o erro

- Ponto de teste – ponto onde a restrição é verificada
  - Imediato – verifica a restrição na passagem pela instância
  - Atrasado – verifica a restrição em algum momento depois passagem pela instância

#### **Características gerais:**

- Elaboração de regras – Simplicidade em elaborar regras e restrições nesta linguagem
  - Simples – é fácil começar a utilizar a linguagem e descrever as regras
  - Média – é necessário um maior conhecimento da linguagem para elaborar as regras
  - Complicada – é necessário um elevado conhecimento da linguagem para elaborar regras
- Integração com o XSD – Se as regras podem ou não ser integradas com nos esquemas XSD
- Ferramentas de apoio – Se existem muitas ou não ferramentas para usar ou aplicar a linguagem
- Organização – a que organização pertence a norma

No quadro seguinte é apresentado o resumo da análise efectuada a cada uma das linguagens em função dos parâmetros definidos. Tendo em conta esta análise, a linguagem escolhida para representar as regras e restrições descritas em EXPRESS, foi a Schematron. A escolha prende-se sobretudo com o facto de o Schematron ser uma linguagem de fácil implementação, existirem muitas aplicações disponibilizadas para usar a linguagem e poder ser integrado com XSD.

<b>Linguagens</b> <b>Parâmetros</b>	<b>Schematron</b>	<b>XCML</b>	<b>XCSL</b>
<b>Valor</b>	Estática	Dinâmica	Dinâmica
<b>Escala</b>	Repositório	Repositório	Repositório
<b>Expressões booleanas</b>	Composta	Composta	Composta
<b>Regras condicionais</b>	Simple	Simple	Composta
<b>Ação</b>	Informativa	Informativa	Informativa
<b>Ponto de teste</b>	Imediata	Imediata	Imediata
<b>Elaboração de regras</b>	Simple	Média	Média
<b>Integração com o XSD</b>	Sim	Não	Não
<b>Ferramentas de apoio</b>	Muitas	Poucas	Poucas
<b>Organização</b>	ISO/IEC	PHILIPS (INTELLECTUAL PROPERTY & STANDARDS)	Univ. Minho

Tabela 2.1 – Quadro de comparações entre as linguagens

A desvantagem que o Schematron apresenta em relação às outras linguagens, isto é, o não suporte de regras dinâmicas, pode ser colmatada estendendo os esquemas com o uso de código Java e XSL, como é possível ver na próxima secção.

### **2.3. Transformação de EXPRESS para Schematron**

Como se analisou, a linguagem Schematron permite definir as regras e restrições presentes em EXPRESS que o XSD não permite. Porém, ainda assim nem todos estes componentes que compõem o EXPRESS são passíveis de ser mapeados para Schematron, como é o caso das funções e procedimentos.

Assim, para tornar possível que os modelos STEP sejam transformados por completo para uma tecnologia capaz de validar dados XML, este trabalho de dissertação, para além de definir como mapear as regras e restrições presentes nos modelos STEP para Schematron, estendeu a linguagem Schematron com funções Java e XSLT.

### 2.3.1. Mapeamento dos modelos STEP

Para conseguir a transformação automática de modelos descritos em EXPRESS para Schematron e Java, é necessário definir padrões de mapeamento, isto é, identificar como cada elemento desta deve ser representado em Schematron ou em Java.

Nesta secção são apresentados alguns exemplos de transformações e os princípios básicos de funcionamento do Schematron. Para complementar esta informação, no Anexo 1 está descrita a tabela de mapeamento, descrevendo como cada elemento de EXPRESS é transformado para Schematron.

Na figura seguinte é ilustrado um exemplo com uma entidade descrita em EXPRESS anteriormente descrita (“automovel”), onde é possível encontrar três regras, uma *UNIQUE* e duas *WHERE*. O primeiro exemplo de transformação diz respeito à regra *UNIQUE*, que como se viu anteriormente, tem o objectivo de garantir que só existe uma “matricula” para cada automóvel. Assim a regra Schematron, tem de procurar no ficheiro XML todas as ocorrências da entidade “automovel” e verificar que o seu atributo “matricula” é único.

```
ENTITY automovel;
  marca: STRING;
  modelo: STRING;
  cor: SET [1:?] OF cores;
  matricula: STRING;
  ano: INTEGER;
  lugares: INTEGER;
  preco: INTEGER;
  iva: INTEGER;
  UNIQUE
    ur1: matricula;
  WHERE
    wr1: { 1885 <= ano <= 2007 };
    wr2: iva = 21;
END_ENTITY;
```

Figura 2.8 – Exemplo da regra UNIQUE e WHERE em EXPRESS

Na figura seguinte é ilustrada a regra transformada em Schematron que permite executar o UNIQUE definido no EXPRESS. Como se pode verificar, o Schematron, tal como foi descrito anteriormente é um ficheiro XML, com elementos “*pattern*”. Assim, para cada regra ou restrição é criado um destes elementos “*pattern*” e cujo nome (“*pattern name*”) ficou definido ser constituído pelo nome da entidade, a que a regra ou restrição diz respeito, e o pelo nome da respectiva regra, neste caso “automovel – ur1”.

```
</pattern>
<pattern name="automovel - ur1">
  <rule context="automovel">
    <report test="not(count(//automovel/matricula[. = current()/matricula]) &lt;= 1) and
not(@ref)">ERROR UNIQUE matricula</report>
  </rule>
</pattern>
```

Figura 2.9 – Exemplo da regra UNIQUE em Schematron

No campo “*rule context*” é introduzido o nome da entidade que se pretende validar, neste caso “automovel” e no campo “*report test*” é introduzida a regra. Neste caso, a regra verifica se o valor do atributo “matricula”, na entidade “automovel”, não se repete ao longo do documento (“not(count(//automovel/matricula[. = current()/matricula]) &lt;= 1) and not(@ref)”). À frente da regra, é definida a mensagem de erro que se pretende que apareça quando esta condição não se verifique (“ERROR UNIQUE matricula”).

Em relação à primeira regra *WHERE*, tem o objectivo de garantir que o ano do automóvel esteja compreendido entre 1885 e 2007. O Schematron equivalente para esta regra está ilustrado na figura seguinte.

```
<pattern name="automovel - WHERE">
  <rule context="automovel">
    <assert test="(ano > 1885 and ano < 2007) or @ref">ERROR WR1
ano</assert>
  </rule>
</pattern>
```

Figura 2.10 – Exemplo da regra WHERE em Schematron

Tal como no exemplo anterior, o campo “pattern name” contém o nome da regra, o “rule context” contém o nome da entidade que se pretende analisar e no campo de teste, desta vez “assert test”, está a regra que se pretende verificar.

A diferença entre o uso *report* e o *assert* é que no primeiro são reportados todos os elementos que verifiquem a condição, enquanto no segundo só são identificados os que não cumprirem a condição de teste.

Por conveniência na explicação, foram utilizados exemplos simples, visto que a transformação de regras e restrições pode torna-se mais complicada, quando por exemplo os atributos não são de tipos simples, sendo estes por exemplo uma referência para outra entidade. Nestes casos a *pattern* tornar-se-á bastante mais complicada, sendo necessário por vezes usar funções Java ou XSL inclusivamente.

Em relação às funções e procedimentos, como exemplo das transformações efectuadas nestes casos, na figura seguinte é ilustrada uma função que se pode encontrar em EXPRESS. Na figura está representada uma função para identificar se um ano é bissexto ou não.

```
FUNCTION BISSEXTO(Ano : INTEGER): BOOLEAN;  
  IF (((Ano MOD 4) = 0) AND ((Ano MOD 100) <> 0)) OR  
      ((Ano MOD 400) = 0)) THEN  
    RETURN(TRUE);  
  ELSE  
    RETURN(FALSE);  
  END_IF;  
END_FUNCTION;
```

Figura 2.11 – Exemplo de uma função em EXPRESS

Como foi dito anteriormente, as funções e os procedimentos não podem ser mapeados para Schematron, sendo assim, é necessário transformá-los em funções Java, podendo depois ser invocadas pelo Schematron. Posteriormente, estas funções são importadas pelos

esquemas de Schematron e utilizadas nas regras e restrições como uma simples função. Na figura seguinte é ilustrado como ficaria em Java a função acima exemplificada.

```
public boolean BISSEXTO(Node Ano) throws Exception {
    try {
        if (Integer.parseInt(Ano.getFirstChild().getNodeValue()) % (4) == 0
            && (Integer.parseInt(Ano.getFirstChild().getNodeValue()) % (100) != 0)
            || (Integer.parseInt(Ano.getFirstChild().getNodeValue()) % (400) == 0))
            {return true;}
        else
            {return false;}
    } catch (Exception e)
        {return false;}
}
```

Figura 2.12 – Função transformada para Java

No anexo 2 é ilustrado um exemplo completo de um modelo em EXPRESS e a sua transformação para Schematron e Java. No exemplo, para além de ser possível ver a complexidade que as regras podem chegar a atingir, é possível ver como as funções Java e as XSLT são integradas com Schematron.

#### **2.4. Arquitectura da ferramenta para mapeamento de STEP-XML**

Para conseguir validar dados XML contra modelos descritos originalmente em EXPRESS, como é o caso dos AP, é necessário transformá-los em XSD e em Schematron (estendido com funções Java e funções matemáticas XSL). Assim, para conseguir este objectivo foi desenhada uma ferramenta de transformação de modelos STEP.

A sua arquitectura, que se pode observar na figura seguinte, está dividida em duas partes, sendo que a primeira é a transformação de EXPRESS (parte sintáctica) para XSD e a segunda a transformação de EXPRESS (parte semântica) para Schematron [40].

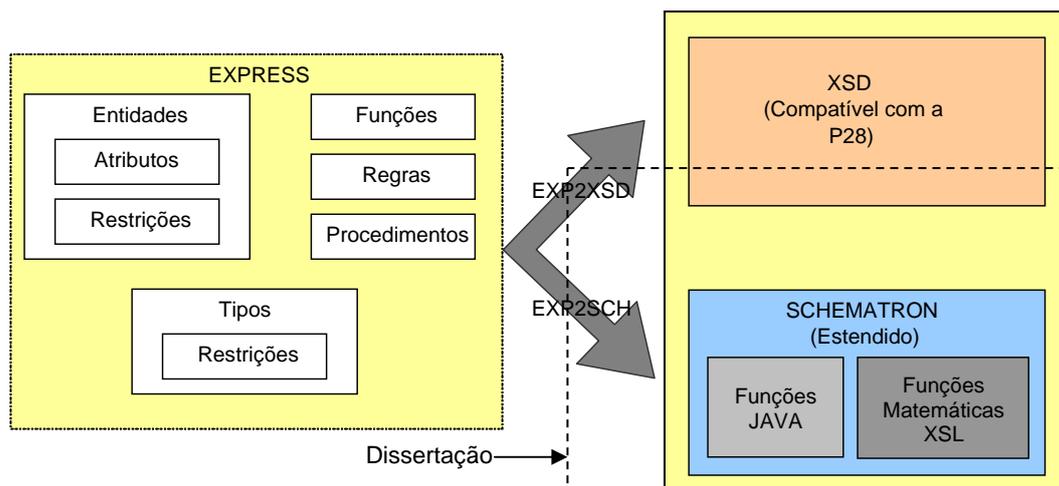


Figura 2.13 – Arquitetura proposta para transformação de modelos STEP.

A ferramenta de transformação de EXPRESS para XSD (EXP2XSD), feita respeitando o definido pela ISO na parte 28, usada nesta arquitetura, não foi elaborada neste trabalho, tendo sido integrada para conseguir obter o XSD necessário para validação dos dados XML.

A segunda ferramenta de transformação (EXP2SCH) desenvolvida neste trabalho de dissertação, permite transformar as partes semânticas descritas nos modelos EXPRESS para Schematron estendido com as funções Java e as funções matemáticas do XSL, desenvolvida neste trabalho, usando como referência a tabela de mapeamentos enunciada na secção anterior.

Deste modo, com o uso destas duas ferramentas, a implementação da arquitetura proposta permite transformar um modelo originalmente descrito em EXPRESS num um modelo equivalente dividido em duas partes, em XSD e Schematron, sendo que estas duas partes conseguem representar por completo toda a informação que é possível encontrar no modelo original.

A execução desta arquitetura possibilita a obtenção do modelo STEP transformado de maneira a poder ser usado numa plataforma para testes de conformidade, como se pode ver nos capítulos seguintes, conseguindo assim superar uma barreira até agora existente para testes de conformidade de dados XML contra os seus modelos definidos em STEP.

### **3. Metodologias e plataformas para testes de conformidade e verificação de interoperabilidade**

#### **3.1. Metodologias e Plataformas existentes**

Os testes de conformidade têm vindo cada vez mais a serem aplicados em conjunto com a aplicação de normas. Assim, muitas organizações para além de definirem as normas elaboram ou sugerem também metodologias para aplicar testes de conformidade.

Também no caso da verificação de interoperabilidade, as organizações têm vindo a denotar uma maior preocupação com este facto, proporcionando metodologias para verificação de interoperabilidade, como complemento às de testes de conformidade.

Seguidamente, são apresentadas metodologias para aplicação de testes de conformidade sobre normas e para verificação de interoperabilidade entre sistemas.

#### **3.1.1. ISO 9646 - “OSI Conformance Testing Methodology and Framework”**

O termo *Open Systems Interconnection (OSI)* é designado para qualificar normas que são usadas para troca de informação entre sistemas “abertos”. Assim, encontrar-se a norma ISO/IEC 7498-1: “*OSI Reference Model for Open Systems*”, que contém um Modelo Referencial cujo propósito é disponibilizar uma base comum para desenvolvimento de normas e permitir a intercomunicação de sistemas [41].

Paralelamente a este modelo referencial foi criada a norma ISO 9646: “*Conformance Testing Methodology and Framework*”, composta por sete partes e que foi originalmente desenvolvida para disponibilizar uma plataforma e definir uma terminologia de testes comuns para sistemas OSI [42]. Devido ao pouco uso dos modelos OSI, a metodologia tem sido pouco utilizada para testes de conformidade a sistemas OSI, não obstante, a metodologia tem sido aplicada a outros tipos de protocolos e sistemas [43].

A metodologia para testes de conformidade, descrita na norma ISO 9646, divide-se em 3 fases como se pode observar na figura seguinte. Na primeira fase da metodologia são definidos os propósitos dos testes, estes propósitos são a base para definir os ATC (*Abstract Test case*<sup>9</sup>) a serem aplicados. A este conjunto de testes dá-se o nome de ATS (*Abstract Test Suite*<sup>10</sup>) e são descritos independentes de qualquer implementação. São descritos utilizando a notação para testes TTCN (*Tree and Tabular Combined Notation*) definido na parte 3 da norma ISO 9646 [44].

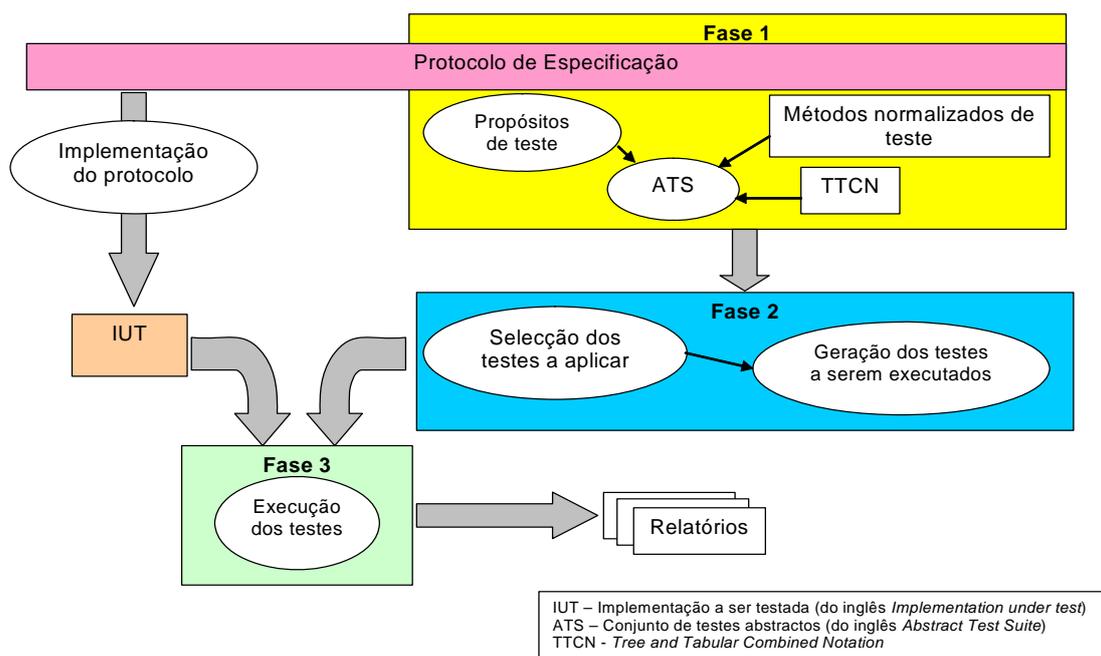


Figura 3.1 – Diagrama adaptado das fases da metodologia para CT da norma ISO 9646.

A segunda fase consiste em seleccionar, de entre todos os testes descritos, quais os testes efectuar. Com os testes seleccionados é necessário transformá-los em testes capazes de serem executados, isto é, capazes de testar o sistema ou equipamento (SUT – *System Under Test*<sup>11</sup>) ao qual se pretende assegurar a sua conformidade.

A ultima fase é a execução dos testes sobre o IUT (*Implementation Under Test*<sup>12</sup>) observando depois qual o seu comportamento perante os testes. O resultado destes é

<sup>9</sup> Casos de teste abstractos

<sup>10</sup> Conjunto de testes abstractos

<sup>11</sup> Sistema testado

<sup>12</sup> Implementação em teste

descrito no PCTR (*Protocol Test Report*<sup>13</sup>), que descreve se a implementação segue os requisitos especificados de conformidade [43].

### **3.1.2. INCITS 1703-D, Conformance Testing Methodology for ANSI/INCITS 358-2002, BioAPI Specification**

A biometria é o estudo estatístico das características físicas ou comportamentais dos seres vivos. Assim, um sistema biométrico pode basear o seu funcionamento em características de diversas partes do corpo humano, por exemplo: os olhos, a palma da mão, as impressões digitais do dedo, a retina ou íris dos olhos [45].

Partindo do princípio de que todas as pessoas são diferentes e têm características distintas, a biometria pode funcionar como uma forma de identificar pessoas, medindo características físicas e comparando-as com dados guardados. Devido ao crescente uso da Biometria, com o objectivo de substituir as palavras-chave, ou garantir uma maior fiabilidade na identificação do indivíduo, têm vindo a ser desenvolvidas diversas APIs (*Application Programming Interface*) Biométricas, denominadas BioAPIs.

O objectivo destas BioAPIs é assegurar o intercâmbio entre componentes de sistemas biométricos, usar uma linguagem comum entre estes sistemas e haver uma independência do utilizador das implementações [46].

Com vista normalizar e especificar estas BioAPIs, foi então criado o consórcio BioAPI, surgindo dele a norma INCITS 358-2002 (*The BioAPI Specification*). Posteriormente em cooperação com a ISO/IEC surgiu a norma internacional ISO/IEC 19784 (*Biometric Application Program Interface (BioAPI) - Part 1 & 2*).

Paralelamente foram criadas duas outras normas, INCITS 1703-D (*Conformance Testing Methodology for ANSI/INCITS 358-2002, BioAPI Specification*) e ISO/IEC 24709

---

<sup>13</sup> Protocolo de relatório dos testes de conformidade

(*Conformance Testing for BioAPI - Part 1 & 2*) cujo objectivo é descrever como efectuar testes de conformidade sobre BioAPIs.

Considerando para este estudo a norma de testes de conformidade ISO/IEC 24709 para BioAPI (visto ser uma norma internacional e conseguir abranger um universo maior de utilizadores), pode-se encontrar na primeira parte desta norma, *Methods and Procedures* (parte 1), uma metodologia para testes de conformidade.

A metodologia descrita visa a aplicação de testes de conformidade em aplicações, plataformas ou serviços provedores de Biometrias (BSP). A metodologia inclui uma descrição completa de uma linguagem para declarações de testes baseada em XML. Esta linguagem porém, está especializada e otimizada para testes de conformidade de BioAPI, não podendo ser usada para programação em geral. Na segunda parte da norma encontram-se descritos nesta linguagem vários testes a serem aplicados para assegurar a conformidade dos sistemas que usam a norma BioAPI [47].

Como é possível ver na figura seguinte os testes descritos são executados pelo BioAPI CTS (*Conformance Test Suite*) que implementa a metodologia de testes de conformidade descritos na norma sobre uma IUT.

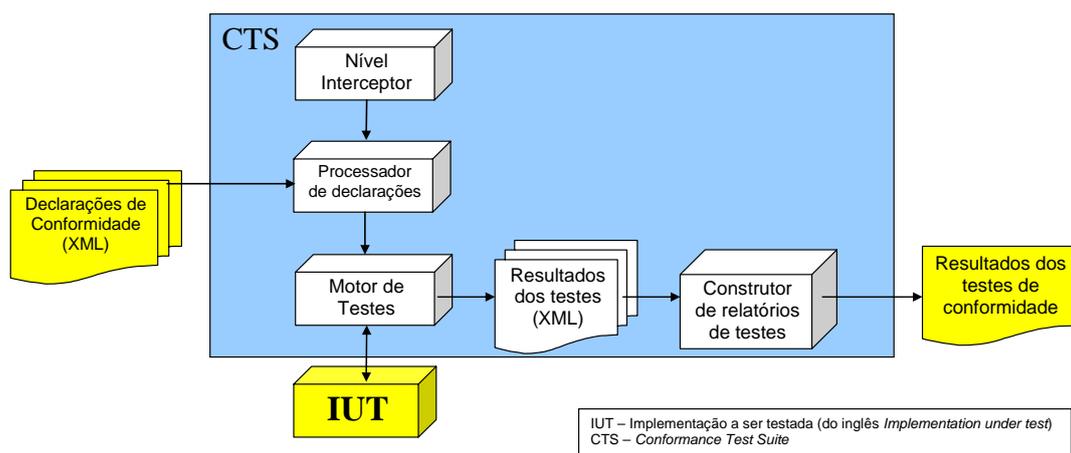


Figura 3.2 – Diagrama adaptado das fases do processo de testes de conformidade para BioAPI.

O CTS começa por transferir as declarações dos testes de conformidade para o processador de declarações. Este processador transforma-as em testes executáveis que são usados pelo

motor de testes para testar a IUT. Os resultados destes testes são descritos em XML e englobados num conjunto de relatórios onde são reportados os resultados dos testes.

### 3.1.3. ISO 10303 parts 30 – *Conformance testing methodology and framework*

A norma STEP (*STandard for the Exchange of Product model data, ISO 10303*) é uma família de normas internacionais cujo objectivo é facilitar o intercâmbio de dados de produto entre sistemas computacionais durante todo o seu ciclo de vida. Esta norma está dividida em várias de partes, entre as quais, nas partes 30 se pode encontrar uma metodologia e plataforma para aplicação de testes de conformidade. Esta metodologia está baseada na já descrita anteriormente 9646, sobretudo na parte 31, onde são apresentados os conceitos gerais relacionados com os testes de conformidade [48].

Na figura seguinte é ilustrado diagrama geral desta metodologia, onde se pode constatar que é bastante semelhante ao da norma ISO 9646. Os testes de conformidade para a norma STEP, também usam conceitos da norma ISO 9646 como ATS, SUT, etc., no entanto na parte da execução dos testes, esta norma faz a distinção entre dois tipos de testes a aplicar, os pré-processados e pós-processados.

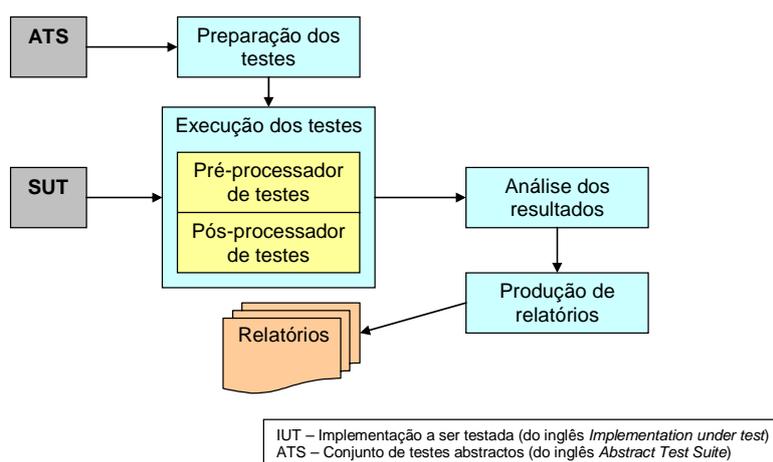


Figura 3.3 – Diagrama adaptado da metodologia usada para testes de conformidade de dados STEP.

Nos testes feitos com o pré-processador começa-se por introduzir o modelo de referência na aplicação para que esta produza dados correspondentes a este. A saída é comparada com

os resultados esperados. No pós-processador é usado um exemplo de dados como entrada do IUT, sendo depois feitas inferências sobre como este exemplo é tratado de modo a concluir se está a ser interpretado de maneira correcta.

### **3.1.4. ETS 300 406 - Methods for Testing and Specification (MTS)**

A *European Telecommunications Standards Institute* (ETSI) tem vindo a desenvolver um conjunto de especificações e normas no sector das telecomunicações e domínios associados. A normalização na área das telecomunicações permite atingir um objectivo desejado por todos nessa área, a interoperabilidade [49].

Com vista a certificar a correcta aplicação destas normas, foi desenvolvida pela ETSI uma norma para descrever uma metodologia de aplicação de testes de conformidade. A Norma ETS 300 406 (*Methods for Testing and Specification (MTS), Protocol and profile conformance testing specifications, Standardization methodology*) descreve essa metodologia normalizada para protocolos e perfis com especificações para testes de conformidade [50].

Os testes de conformidade da norma ETS 300 406 usam princípios básicos da norma anteriormente descrita, a ISO 9646. Como se pode ver na figura seguinte, nesta metodologia começa-se por identificar os propósitos de teste (TP<sup>14</sup>) e a arquitectura/métodos dos testes abstractos (TSS<sup>15</sup>) a partir da norma ou protocolo à qual se pretende efectuar os testes de conformidade [50]. Posteriormente, baseados nos TP e TSS, são definidos os ATC descritos em TTCN e com estes é definido o ATS.

TTCN, como se viu anteriormente, é uma linguagem usada para descrever casos de testes e foi desenvolvida pela ISO e publicada na norma ISO 9646-3. Esta linguagem tem vindo a ser bastante usada no desenvolvimento de ATS por parte de algumas tecnologias como a GSM, UMTS, DECT, ISDN, etc [51].

---

<sup>14</sup> Do termo inglês *Test Purposes*

<sup>15</sup> Do termo inglês *Test Suite Structure*

Devido ao seu grande uso e necessidade de adaptação a novos requisitos, a ETSI em 2001, redesenhou a linguagem e publicou-a com um novo nome, *Testing and Test Control Notation* (TTCN-3) [52]. Esta nova versão permite a geração de testes executáveis automaticamente. Os testes depois de definidos em TTCN-3 são compilados, como por exemplo, para Java, C ou C++ [55]. Estes testes são executados sobre a implementação que usa a norma à qual se quer efectuar os testes de conformidade. Como resultado dos testes é produzido um conjunto de relatórios onde são descritos os erros encontrados na aplicação dos testes [50].

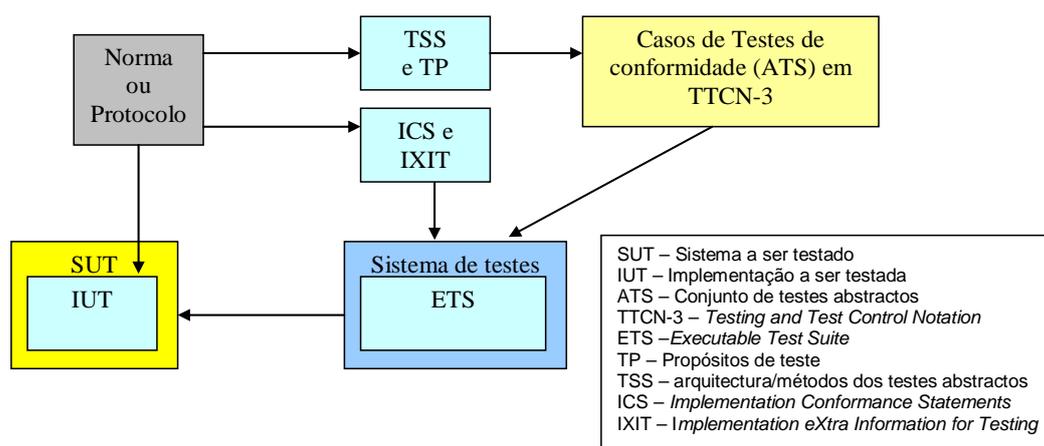


Figura 3.4 – Diagrama adaptado da metodologia de testes de conformidade da ETSI.

Paralelamente, também baseado na norma ou protocolo, é definida a lista das capacidades, funcionalidades, opções e características da IUT. Esta lista é chamada de ICS (*Implementation Conformance Statements*<sup>16</sup>) e pode ser usada para seleccionar e parametrizar os ATC. Também usado o efeito de seleccionar e parametrizar, as normas ou protocolos, o IXIT (*Implementation eXtra Information for Testing*<sup>17</sup>) reúne informações adicionais, como endereços os valores de tempo [53].

### 3.1.5. TS 102 237-1; Interoperability test methods & approaches

Embora seja possível encontrar algumas metodologias para testes de conformidade, a nível de metodologias para verificação de interoperabilidade já não acontece o mesmo, assim, com vista a suprir esta lacuna de metodologias de verificação de interoperabilidade, a ETSI EP TIPHON (*Telecommunications and Internet Protocol Harmonisation Over Networks*)

<sup>16</sup> Declarações de conformidade para a implementação

<sup>17</sup> Informações extra para a implementação em teste

descreve na norma TS 102 237-1 "Interoperability Test Methods & Approaches; Part 1: Generic Approach to Interoperability Testing" uma metodologia genérica para testes de interoperabilidade em sistemas de comunicação [51].

Publicada em 2003, esta norma ainda está em fase de desenvolvimento, podendo ainda sofrer algumas alterações, no entanto, com a cooperação com ETSI TC MTS (*Methods for Testing and Specification*), a ETSI espera que esta seja aceite em toda a sua comunidade.

Esta metodologia é baseada na norma ISO/IEC 9646 para testes de conformidade, usando muitos dos conceitos da norma ISO/IEC 9646, mas adaptados para interoperabilidade. As principais diferenças, como se pode ver na figura seguinte, são que o SUT que em vez de ter um IUT passa a conter um EUT (*Equipment Under Test*<sup>18</sup>) e um ou mais RE (*Reference Equipment*<sup>19</sup>).

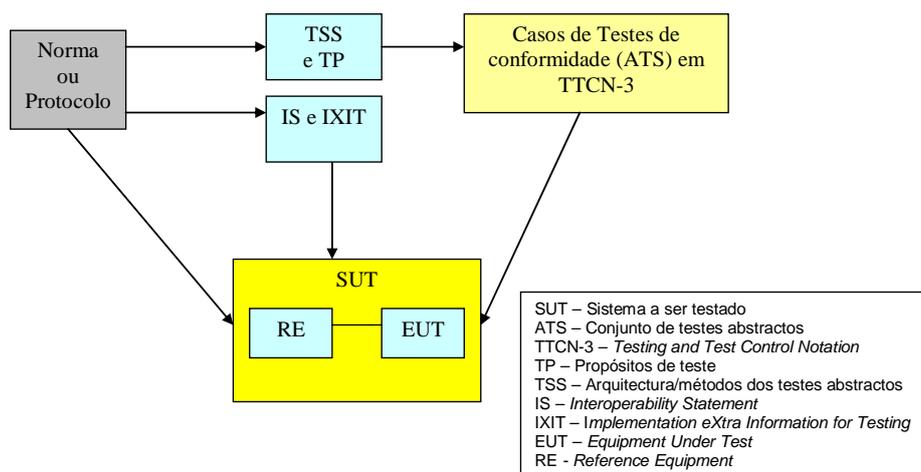


Figura 3.5 – Diagrama adaptado da metodologia de verificação de interoperabilidade da ETSI.

Assim, neste caso os TP e os TSS, também derivados das normas, são identificados para testar funcionalidades específicas da EUT. Outra diferença é que em vez de ICS, é definida agora uma lista de capacidades e funcionalidades para o EUT chamada de IS (*Interoperability Statement*<sup>20</sup>) [53]. A outra grande diferença é que neste caso pode não ser necessário a existência de ETS, visto só ser necessário se forem definidos *scripts* nos ATS.

<sup>18</sup> Equipamento a ser testado

<sup>19</sup> Equipamento de referência

<sup>20</sup> Declarações de interoperabilidade

Caso isto aconteça, pode ser utilizado a TTCN-3 novamente para assim gerar os testes executáveis a partir dela.

### **3.2. Análise comparativa das metodologias**

As metodologias estudadas, embora descritas para diferentes normas, têm aspectos bastante semelhantes, sendo esta semelhança sobretudo evidenciada quando se constata que muitas delas se baseiam na norma ISO 9646. Outra semelhança em algumas nas metodologias, embora com diferentes versões, é o uso da linguagem TTCN para descrição dos ATC. O uso desta linguagem é compreensível visto facilitar bastante a descrição dos ATC e no caso do TTCN-3 proporcionar inclusive a geração automática de testes executáveis.

Tendo em conta o objectivo desta dissertação e para estruturar a análise das metodologias, foram definidos vários critérios de avaliação:

- Execução de pré e pós testes sobre o IUT;
- Qual a linguagem usada para descrever os testes abstractos;
- Facilidade de implementação da metodologia;
- Validação dos dados contra o seu modelo conceptual;
- Geração automática de testes executáveis a partir dos ATS;
- Adaptabilidade a diferentes modelos de dados.

Utilizando estes critérios, foram analisadas as metodologias descritas anteriormente, obtendo a tabela seguinte. Na tabela, para além das metodologias estudadas é possível encontrar uma nova metodologia identificada como “CT2XML”, que representa a metodologia tida como ideal para implementação neste trabalho de dissertação, isto é, para a validação de dados XML contra modelos conceptuais STEP.

<b>Metodologias</b>	<b>ISO 9646</b>	<b>INCITS 1703-D</b>	<b>ISO 10303 (Partes 30)</b>	<b>ETS 300 406</b>	<b>CT2XML</b>
<b>Pré e Pós testes</b>	×	×	✓	×	×
<b>Definição dos ATS</b>	TTCN	XML	Texto	TTCN-3	TTCN-3
<b>Dificuldade de implementação</b>	±	↑	↑	±	↓
<b>Validação do Modelo de dados (Sintaxe e semântica)</b>	×	×	×	×	✓
<b>Geração automática de testes executáveis</b>	×	✓	×	✓	✓
<b>Adaptabilidade a diferentes modelos</b>	±	↓	±	±	↑

(legenda: ✓ - sim; × - não; ↑ - elevada; ↓ - Baixa; ± - média)

Tabela 3.1 – Quadro de comparações entre as metodologias

Assim, as principais características desta metodologia CT2XML são:

- Não necessitar de ter pré e pós testes, visto os testes que se pretendem efectuar serem todos sobre ficheiros gerados pelas aplicações;
- Utilizar o TTCN-3 para descrever os ATC, facilitando assim a geração de testes executáveis;
- Ser fácil de implementar;
- Efectuar validação sintáctica e semântica dos dados em XML;
- Ser fácil de adaptar a novos modelos, isto é, conseguir efectuar testes de conformidade a vários tipos de modelos e não só para os quais foi projectada.

Após o estudo das várias metodologias, pode-se concluir que nenhuma destas reúne todos os requisitos identificados na CT2XML. De ressaltar o facto desta metodologia não ter como essencial o facto de existir pré e pós testes, devendo-se este facto a que a metodologia

pretendida venha apenas implementar os pós testes, isto é, validar os ficheiros resultantes das implementações.

A metodologia de verificação de interoperabilidade não foi incluída na tabela visto possuir as mesmas características que a outra norma, ETS 300 406, proposta pela ETSI, e que não é uma metodologia de testes de conformidade, sendo esta a única metodologia de verificação de interoperabilidade.

Em conclusão e tendo em consideração os requisitos do mercado de hoje em dia, é proposta no capítulo seguinte uma metodologia, que suporta as características identificadas para a CT2XML, acima identificada. Esta metodologia pretende suprir as lacunas existentes nas outras metodologias, no caso particular desta dissertação.

## **4. Testes de conformidade e verificação de interoperabilidade para protocolos de aplicação STEP representados em XML**

Com vista a conseguir validar dados em XML contra protocolos de aplicação STEP, neste capítulo é apresentada uma metodologia para testes de conformidade que visa cumprir com os critérios enunciados no capítulo anterior. É também apresentada uma plataforma que implementa esta mesma metodologia.

Neste mesmo capítulo, é proposta também uma metodologia e plataforma para verificação de interoperabilidade para sistemas que usem protocolos STEP representados em XML. Esta metodologia, juntamente com a de testes de conformidade, forma parte de uma proposta para uma metodologia integrada que permite o teste completo a um sistema em termos de conformidade e interoperabilidade. Tal como para as outras metodologias, é também apresentada uma plataforma para implementação desta metodologia integrada.

### ***4.1. Metodologia para testes de conformidade***

Existem várias metodologias para aplicação de testes de conformidade, mas estas têm algumas lacunas como foi identificado no capítulo anterior. Nesta secção, é proposta uma nova metodologia, que usando algumas partes das metodologias estudadas, soluciona as lacunas identificadas.

A metodologia descrita é baseada sobretudo na norma ISO 10303 partes 30, acrescentando partes como a validação da estrutura de dados, tanto sintacticamente como semanticamente, adicionando o TTCN-3 para descrever os ATS e simplificando partes como a dos pré e pós testes.

Esta metodologia prevê dois tipos de testes de conformidade, os testes básicos, que fazem os testes preliminares para conformidade do IUT e os testes de capacidade que testam as

capacidades do IUT contra as descritas no PICS (*Protocol Implementation Conformance Statement*).

Estes dois tipos de testes permitem satisfazer dois tipos de utilizadores, os que apenas pretendem verificar se os seus dados estão em conformidade com a sintaxe e semântica de um qualquer modelo conceptual STEP e os utilizadores que pretendam confirmar também características descritas em PICS contidas nos AP ou outros casos de testes definidos pelo próprios utilizadores.

Assim, os testes básicos contemplam os testes de conformidade dos dados, em termos sintácticos e semânticos, contra o seu modelo, definido em STEP e os testes de capacidade, fazem a validação dos ATS, tendo estes as duas origens enunciadas anteriormente. Na metodologia descrita, os dois testes estão englobados, sendo que os testes de conformidade ficam preparados para executar qualquer dos dois testes. “Qual a usar”, é o utilizador da metodologia que tem de escolher [54].

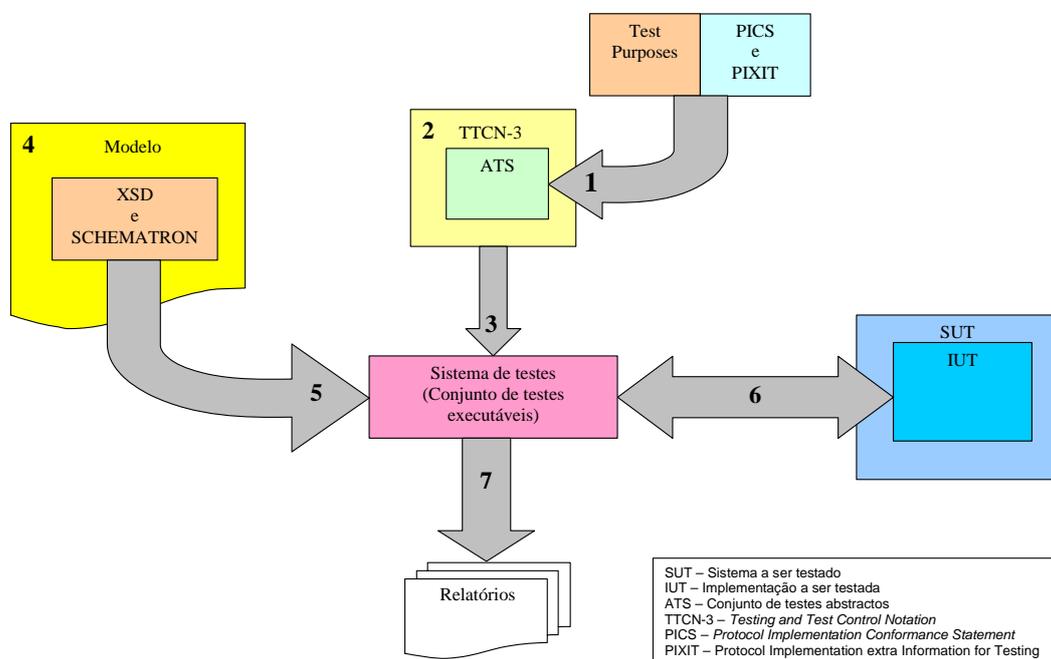


Figura 4.1 – Diagrama da metodologia proposta de testes de conformidade

Na figura anterior é possível ver o diagrama da metodologia proposta, onde se pode identificar como primeiro passo desta (nº 1 na figura), a definição do ATS. O ATS é o conjunto dos ATC, definidos a partir dos propósitos do teste, dos PICS e os PIXIT (*Protocol Implementation extra Information for Testing*).

O PICS é o conjunto das opções definidas para um sistema de testes. Estas opções são seguidas pelo sistema de testes de conformidade para poder concluir se os requisitos da norma estão a ser cumpridos ou não pela implementação. Sendo assim, cada norma fornece sempre um questionário onde são descritas as capacidades que a implementação deve cumprir para estar em conformidade com a norma.

O PIXIT contém a informação sobre o IUT que está a ser testado, isto é, através do PIXIT é possível saber quais os testes mais apropriados a serem efectuados para o IUT ser analisado, tendo em conta o PICS e outras informações administrativas [55].

Neste primeiro passo, estes testes podem ser definidos textualmente sem usar nenhuma linguagem específica, isto é, podem ser descritos em português ou inglês em frases simples. Após a obtenção do ATS, este deve ser descrito numa linguagem capaz de transformar estes testes em testes executáveis.

Deste modo, o segundo passo (nº 2 na figura) desta metodologia, é a descrição dos ATC definidos anteriormente, usando a linguagem TTCN-3. Esta descrição pode ser feita através de um editor de texto ou um editor gráfico TTCN-3. A descrição dos ATC em TTCN-3 permite a geração automática de testes executáveis para várias tecnologias.

Como é ilustrado na figura anterior, após ter todos os ATC descritos em TTCN-3, é necessário transformá-los em testes executáveis. Sendo assim, o passo três é a geração destes mesmos testes executáveis, que são usados posteriormente para testar as implementações nas quais se quer efectuar os testes de conformidade [52].

Como é possível observar no diagrama da metodologia, em paralelo com estes passos é feita também a transformação do modelo STEP (descrito em EXPRESS) para os seus esquemas equivalentes em XSD (XML *Schema*) e Schematron (nº4 na figura). Para o XSD, como foi descrito no capítulo 2, é transformada a parte sintáctica do modelo, isto é, as entidades, atributos e tipos e para Schematron a parte semântica, como as regras *Where*, *Unique* e Funções.

Estes dois esquemas definidos, que constituem a base de conhecimento a usar para efectuar testes de conformidade aos dados em XML, são incorporados na plataforma para execução de testes (passo nº5). Por último, e depois de ter todos os testes a serem efectuados sobre a implementação, os testes, são executados sobre a implementação (passo nº6). Destes testes, são produzidos relatórios onde constam todos os erros encontrados na implementação (passo nº7). Se não existirem erros pode-se então afirmar que a implementação está em conformidade com a norma.

#### **4.2. Plataforma SOA para testes de conformidade**

Com vista a implantar a metodologia descrita, foi criada uma plataforma de suporte à execução de testes de conformidade. Esta plataforma é baseada numa arquitectura orientada a serviços (SOA<sup>21</sup>). As SOA são arquitecturas que colocam o seu foco na disponibilização de serviços independentes, que normalmente são acedidos através da Internet.

A disponibilização destes serviços através da Internet permite a integração em plataformas ou sistemas, ou mesmo em outros serviços. Deste modo, a plataforma para testes de conformidade, ao ficar disponível em forma de serviços, faculta que qualquer implementação possa incorporar estes serviços.

---

<sup>21</sup> Do inglês *Service Oriented Architecture*

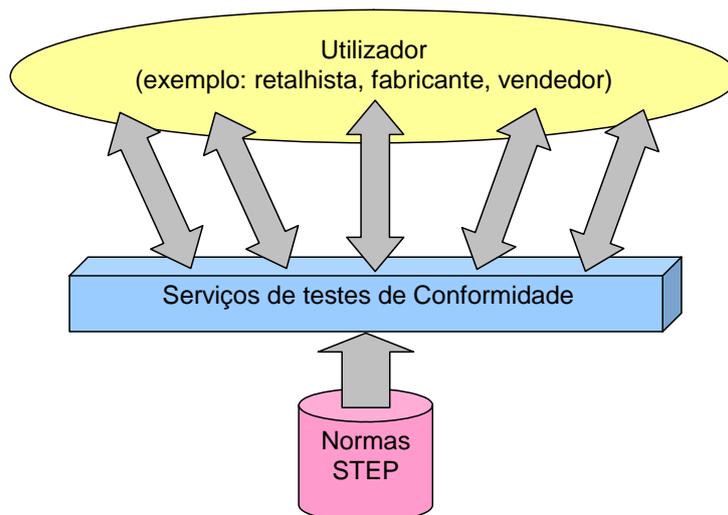


Figura 4.2 – Serviços de testes de conformidade.

É assim possível que as empresas ou outro tipo de interessados, como Universidades, centros de investigação, elementos da comunidade ISO, etc, que usem dados XML e estes obedeçam a normas STEP ou tenham o seu modelo descrito em EXPRESS, possam invocar o serviço e obter como resposta um relatório com o resultado dos testes de conformidade.

Para chegar até à disponibilização dos serviços de testes de conformidade online, existem vários passos a seguir. Um destes passos, como se viu na secção anterior, é a transformação dos modelos descritos em STEP para os seus equivalentes em XSD e Schematron. Estas transformações, já explicadas no capítulo 2, são efectuadas usando as ferramentas também enunciadas nesse mesmo capítulo, EXP2XSD e EXP2SCH.

Na figura seguinte é possível ver um exemplo de uma entidade STEP descrita em EXPRESS e os modelos equivalentes em XSD e Schematron, resultantes da transformação efectuada pela ferramenta de transformação descrita no capítulo 2. A entidade contém também uma regra que utiliza uma função "ValidCalendarDate", que está descrita também no modelo STEP. Esta função é transformada numa função Java, que permite ser invocada pelo Schematron.

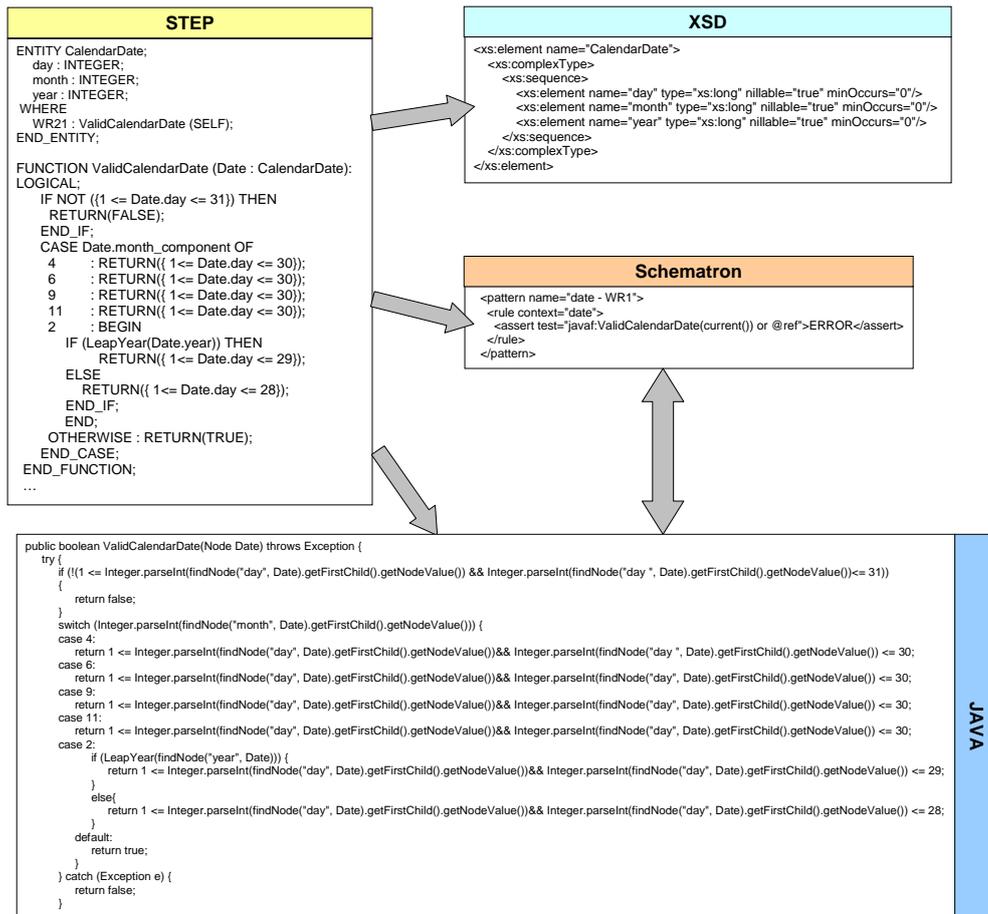


Figura 4.3 – Exemplo de transformação

O conjunto de modelos gerados, XSD, Schematron com ligações para as funções Java e Matemáticas do XSL, constituem a base de conhecimento usada para testar os dados, em termos sintácticos e semânticos. Assim, existem dois testes diferentes no caso de conformidade para com o modelo de dados, o sintáctico e o semântico.

Estes dois testes, como foi descrito na secção anterior, fazem parte dos testes básicos. Deste modo, os testes básicos, são disponibilizados através de um serviço, que contém dois métodos, um para efectuar testes sintácticos e outro testes semânticos. Isto permite que haja uma selecção de qual o teste a implementar, embora só se possa concluir que os dados passaram os testes básicos se forem efectuados os dois testes.

Na validação dos dados contra o seu modelo, é usado um motor de aplicação que usa o Xalan<sup>22</sup>, que por sua vez usa o SAX<sup>23</sup> para fazer o “parser” dos ficheiros, como base para todas transformações e validações. Desenvolvidas em Java, as transformações e as validações, permitem obter o relatório com os erros encontrados.

Para completar os outros tipos de testes, os de capacidade, é necessário transformar o ATS para testes executáveis, sendo assim possível efectuar os testes sobre os ficheiros de dados. Para conseguir este objectivo é usado um compilador de dados TTCN-3 que gera em Java os testes contidos no ATS. O Java foi escolhido visto ser a linguagem usada na plataforma, permitindo assim a fácil inclusão de um ficheiro executável Java (.jar) na plataforma de modo a permitir efectuar os testes presentes no ATS.

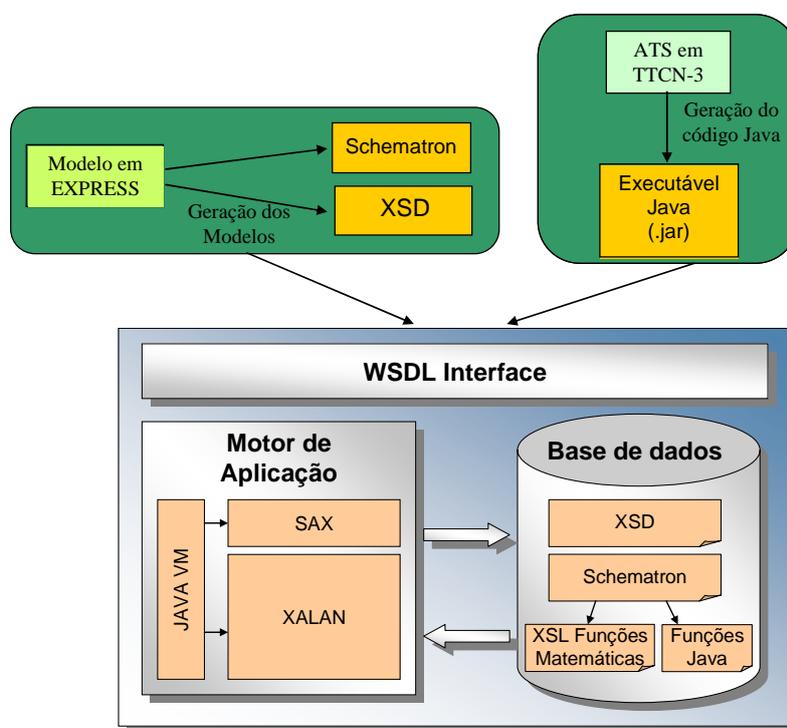


Figura 4.4 – Arquitectura proposta da plataforma de testes de Conformidade.

Na figura anterior é ilustrada a arquitectura até agora descrita, constatando-se que depois de conseguir todos os testes a efectuar sobre os dados XML, são gerados os serviços de testes de conformidade, em WSDL. Na figura seguinte é possível ver um exemplo de WSDL gerado

<sup>22</sup> <http://xml.apache.org/xalan-j/>

<sup>23</sup> <http://www.saxproject.org/>

para testes de conformidade básicos, onde se podem ver os dois métodos enunciados anteriormente, o sintáctico, identificado como “xsd”, e o semântico, identificado como “rules”.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:map="http://systinet.com/mapping/" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:ns0="http://systinet.com/xsd/SchemaTypes/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://systinet.com/wsdl/FedexCT/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://systinet.com/wsdl/FedexCT/" name="FedexCT.FedexCT">
+ <types>
+ <message name="Exception">
- <message name="FedexCT_rules_Request_Soap">
  <part name="request" type="xsd:base64Binary" />
</message>
- <message name="FedexCT_rules_Response_Soap">
  <part name="response" type="tns:AResponseMessageAttachment" />
</message>
- <message name="FedexCT_xsd_Request_Soap">
  <part name="request" type="xsd:base64Binary" />
</message>
+ <message name="FedexCT_xsd_Response_Soap">
  <part name="response" type="tns:AResponseMessageAttachment" />
</message>
- <portType name="FedexCT">
  - <operation name="rules" parameterOrder="request">
    <input message="tns:FedexCT_rules_Request_Soap" />
    <output message="tns:FedexCT_rules_Response_Soap" />
    <fault name="Exception" message="tns:Exception" />
  </operation>
  - <operation name="xsd" parameterOrder="request">
    <input message="tns:FedexCT_xsd_Request_Soap" />
    <output message="tns:FedexCT_xsd_Response_Soap" />
  </operation>
</portType>
+ <binding name="FedexCT" type="tns:FedexCT">
- <service name="FedexCT">
  - <port name="FedexCT" binding="tns:FedexCT">
    <soap:address location="http://172.16.3.212:6060/FedexCT/" />
  </port>
</service>
</definitions>
```

Figura 4.5 – Exemplo de WSDL gerado.

Disponíveis através da Internet, os serviços, permitem que um utilizador que pretenda validar os seus dados, invoque o serviço especificando o método a usar, enviando em anexo o ficheiro de dados que pretende validar. O ficheiro é validado conforme o pedido efectuado e é enviado de volta, em anexo, o relatório com os erros encontrados. O utilizador analisando o relatório pode assim corrigir os erros para que os seus dados fiquem em conformidade com a norma.

### **4.3. Metodologia e plataforma de verificação de interoperabilidade**

Como foi analisado no capítulo anterior, não existem muitas metodologias de referência para verificação de interoperabilidade. A metodologia estudada, desenvolvida pela ETSI, é usada em sistemas de telecomunicações, sendo bastante parecida com a outra proposta pela ETSI para testes de conformidade.

Como por sua vez, a metodologia descrita neste capítulo, tem a mesma base (ISO 9646) que a apresentada pela ETSI para verificação de interoperabilidade, pode concluir-se que a metodologia a aplicar não será muito diferente.

Neste trabalho, em termos de verificação de interoperabilidade, o objectivo é verificar que dois sistemas que usem dados XML e tenham uma norma ou um modelo STEP para os regulamentar são interoperáveis.

Deste modo, a metodologia proposta para garantir a interoperabilidade destes sistemas é baseada na metodologia proposta neste capítulo para testes de conformidade. A diferença entre estas está na definição dos casos de testes que se pretende aplicar, que têm de ser pré-acordados entre os intervenientes, e em como aplicar a verificação de interoperabilidade.

Os casos de testes para esta metodologia, são definidos de forma a verificar critérios necessários para garantir que os sistemas são interoperáveis em relação à troca de dados XML, i.e., o que é definido no sistema fonte é interpretado exactamente da mesma maneira no sistema destino, e reciprocamente, Tal como nos testes de conformidade, os testes são definidos por especialistas no domínio e podem ser adaptados a cada implementação sendo estes também incluídos no ATS.

Visto a metodologia ser uma extensão à anterior, o diagrama da metodologia não é novamente apresentado, é apenas ilustrada na figura seguinte a forma como a metodologia deve ser aplicada no caso da verificação de interoperabilidade.

Assim, como é possível ver na figura, o primeiro passo na aplicação da metodologia, é garantir que os dados que vão ser enviados estejam em conformidade com o seu modelo e além disso cumpram as especificidades descritas nos ATC. Depois de receber o relatório (representado pelo número dois na figura) o utilizador consulta-o, vendo se os dados gerados pelo seu sistema são interoperáveis. Se existirem erros, estes devem ser corrigidos e executados novamente os testes para verificação de interoperabilidade, se o relatório

confirmar a interoperabilidade do sistema e o outro utilizador também tiver a confirmação de que os seus dados são interoperáveis, podem então iniciar a comunicação (número três na figura).

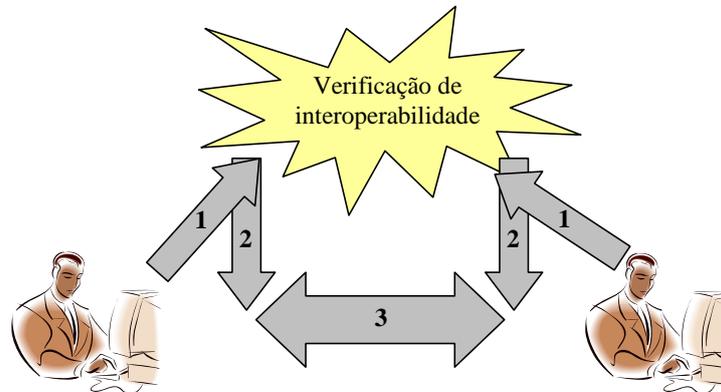


Figura 4.6 – Aplicação da metodologia para verificação de interoperabilidade

Deste modo, a metodologia proposta nesta dissertação para verificação da interoperabilidade para sistemas que usem dados XML baseados em modelos conceptuais STEP, é a execução dos testes de verificação de interoperabilidade por ambas as partes antes de iniciar a comunicação. Sendo completada, esta verificação garante que os dados estão de acordo com o modelo e que os ATC definidos para a comunicação são cumpridos, garantindo assim interoperabilidade.

A arquitectura para implementar a metodologia proposta, tal como no caso dos testes de conformidade, é baseada numa arquitectura SOA, disponibilizando o serviço de maneira a integrá-lo numa implementação já existente. Esta arquitectura, visto ser idêntica à arquitectura descrita para os testes de conformidade, não será novamente aqui apresentada.

#### **4.4. Integração dos testes de conformidade e verificação de interoperabilidade**

As metodologias e as arquitecturas propostas para testes de conformidade e verificação de interoperabilidade são extensões às anteriores, apenas se alterando o conteúdo dos casos de testes. Este facto deve-se sobretudo a que a verificação de interoperabilidade não se

pode, neste caso, dissociar dos testes de conformidade, visto que, se os dados estiverem em conformidade com o modelo e se os dois sistemas utilizarem o mesmo modelo, estes conseguem compreender os dados um do outro, dando assim o primeiro passo para a interoperabilidade.

Tendo em atenção que apenas o ATS é diferente nestas duas metodologias, é possível integra-las numa só tal como a arquitectura para a plataforma. Deste modo, qualquer utilizador que pretenda efectuar testes de conformidade ou verificar a interoperabilidade do seu sistema, ou ambos, pode utilizar o mesmo método para o fazer, isto é, através de um serviço disponível online.

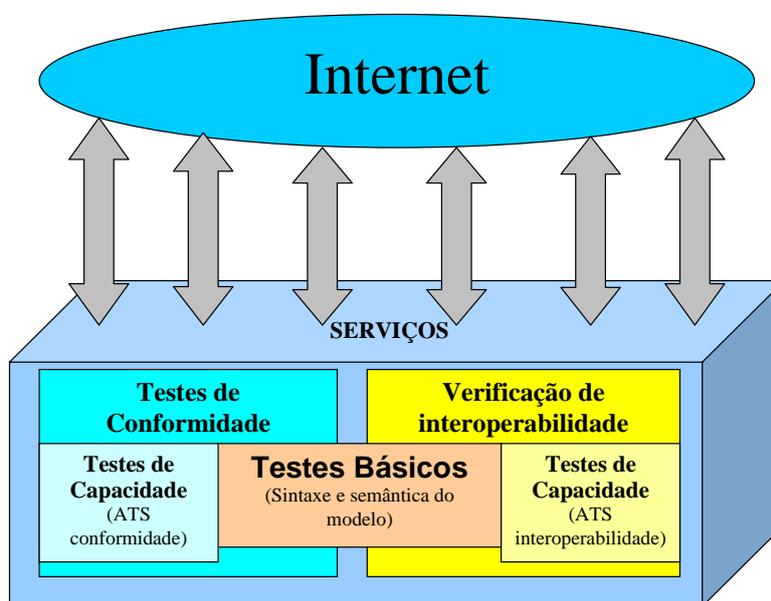


Figura 4.7 – Arquitectura conjunta para testes de conformidade e verificação de interoperabilidade

Como é possível observar na figura, a arquitectura para suportar a metodologia descrita integra as duas arquitecturas, disponibilizando assim dois serviços distintos. Estes serviços são compostos por um método igual, o de conformidade dos dados em termos de sintaxe e semântica do modelo, e por dois métodos diferentes, ATS dos testes de conformidade e ATS da verificação de interoperabilidade.

A implementação desta arquitectura permite disponibilizar através da Internet os serviços necessários para garantir que qualquer empresa, ou outro tipo de utilizador, que utilize dados

XML e em que estes sejam baseados numa norma STEP, possa garantir que estão em conformidade com a norma e ao mesmo tempo que pode trocar informação com outra empresa que utilize a mesma norma para os seus dados XML, garantindo assim a interoperabilidade.

## 5. Validação e demonstração em ambientes industriais

O trabalho desta dissertação esteve envolvido nos projectos Europeus SMART-fm e ATHENA e relacionado com a norma internacional AP236. O envolvimento nestes projectos permitiu a validação do trabalho desenvolvido assim como o desenvolvimento e aperfeiçoamento, da metodologia e da plataforma.

Este trabalho esteve presente, em diferentes fases de desenvolvimento, em vários cenários piloto dentro destes projectos, sobretudo no ATHENA. Neste capítulo são descritos dois cenários, do projecto ATHENA, onde a plataforma de testes de conformidade foi aplicada. A inclusão da plataforma de testes de conformidade nestes cenários permite vislumbrar a grande importância que a utilização de testes de conformidade tem hoje em dia.

É de notar que a parte de verificação de interoperabilidade, não foi aplicada de uma forma directa nos cenários apresentados, isto é, não foram implementados casos de testes para verificar a interoperabilidade. Este facto deve-se à fase embrionária em que o trabalho nesta área ainda se encontra e ao foco principal desta dissertação serem os testes de conformidade.

No entanto, o primeiro passo para garantir a interoperabilidade, como foi possível constatar na metodologia de verificação de interoperabilidade, é a conformidade dos dados com o seu modelo, garantindo que pelo menos à partida não existam erros sintácticos e semânticos.

Os dois cenários apresentados mostram as diferentes aplicações destes testes nas várias áreas, sendo que o primeiro cenário é na área de *e-procurement*<sup>24</sup> no mobiliário e o segundo na área de entrega e distribuição de produtos.

---

<sup>24</sup> Pesquisa online

### 5.1. Cenário de aplicação, e-Procurement

O cenário apresentado nesta secção faz parte do projecto ATHENA, mais concretamente do sub-projecto B5, sendo um dos cenários de validação das ferramentas do projecto. O projecto estava dividido em quatro áreas da indústria: automóvel, mobiliário, aeronáutica e telecomunicações. Cada uma delas teve um cenário proposto, usado para testar a aplicação real das ferramentas, no caso do mobiliário o cenário usado foi o de *e-procurement* [56].

O cenário descreve as interações que existem entre um fabricante e um retalhista até chegarem à troca efectiva de produtos, quando um cliente faz um pedido a esse retalhista. Isto é, o cenário relata os passos que antecedem à entrega de um dado produto, como por exemplo o pedido de orçamento.

Admitindo que todo este processo é feito através da troca de dados pela Internet, o diagrama geral deste cenário é o apresentado na figura seguinte. Na figura é possível constatar que quando um retalhista pretende encomendar um dado produto, o primeiro passo é o pedido de orçamento, ao qual o fabricante responde enviando o orçamento para o produto pretendido.

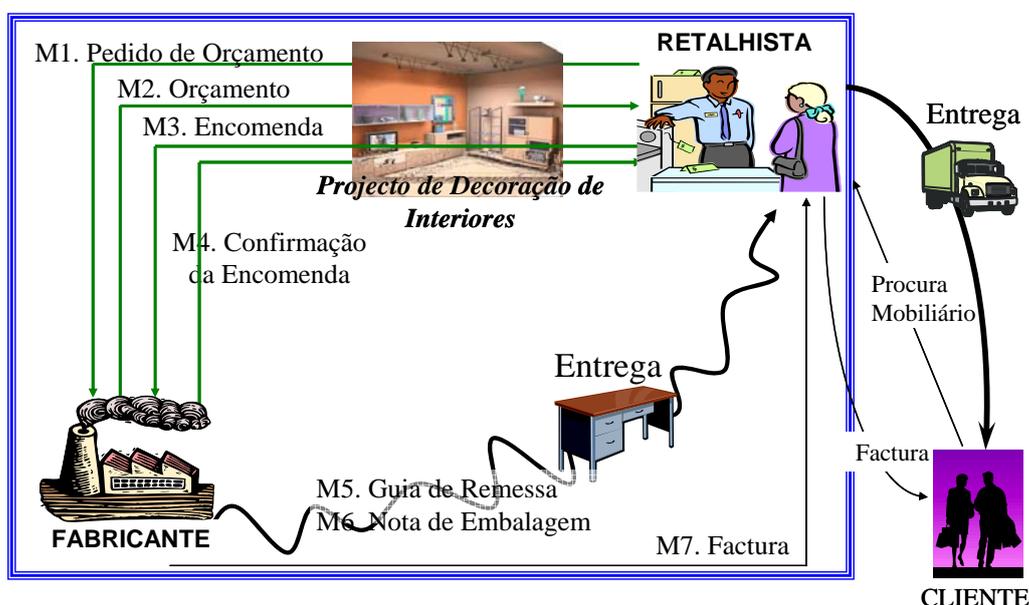


Figura 5.1 – Cenário 1 – e-Procurement.

Se o retalhista concordar com o preço e as condições oferecidas pelo fabricante, envia o pedido de encomenda, à qual o fabricante confirma a sua recepção e disponibilidade em satisfazê-la.

Os dados usados na troca de mensagens entre o fabricante e o retalhista obedecem a um modelo. Este modelo foi descrito em EXPRESS, com vista a estabelecer uma ligação com a norma AP236, visto ser um processo relacionado com o mobiliário. No entanto, os dados usados para troca no cenário estão em XML. Como se viu nesta dissertação, isto coloca um problema quando se pretende efectuar testes para garantir a conformidade dos dados com o seu modelo.

Assim, para garantir que os dados cumprem a norma estabelecida para a troca de mensagens, foi usada a metodologia e plataforma propostas nesta dissertação. Através da implementação dos testes de conformidade é possível detectar os erros nas mensagens trocadas.

Como exemplo para demonstração do cenário, foi criado o serviço de testes de conformidade para o pedido de orçamento. Tal como é descrito na metodologia o primeiro passo foi transformar o modelo STEP para XSD e Schematron e posteriormente criado o serviço com estes dois modelos como base de conhecimento para efectuar os testes.

```
ENTITY Products_information;  
  productRecord : LIST [1:?] OF Product;  
  quantity : NUMBER;  
  WHERE  
    WR1 : VALUE_UNIQUE(productRecord);  
    WR2 : quantity > 0;  
  END_ENTITY;  
  
ENTITY Product;  
  id : STRING;  
  name : STRING;  
  description : OPTIONAL STRING;  
  END_ENTITY;
```

Figura 5.2 – Exemplo de duas entidades do pedido de orçamento.

Na figura anterior estão representadas, como exemplo, em EXPRESS duas das entidades que fazem parte do modelo de um pedido de orçamento. A primeira entidade,

“*Products\_information*”, contém o conjunto de produtos pretendidos, assim como a sua quantidade. Os produtos pretendidos são incluídos numa lista de produtos (“*productRecord*”), sendo cada um identificado através da entidade “*Product*”, que contém os respectivos atributos. Na entidade “*Products\_information*”, existem também duas regras, uma para verificar se não são repetidos produtos na lista de produtos e a segunda que a quantidade de produtos seja superior a zero.

Executando o primeiro passo da metodologia, isto é, transformando o modelo STEP em XSD, é obtido um modelo, como o exemplificado na figura seguinte para as duas entidades do exemplo. É através deste XSD gerado, que são feitos os testes sintáticos sobre todas as instâncias XML destas entidades. Tendo as entidades transformadas em XSD, falta apenas, para completar o primeiro passo, a transformação da parte semântica do modelo, isto é, as regras presentes na entidade “*Products\_information*”.

```

<xs:element name="Products_information" type="Products_information" substitutionGroup="ex:entity"
nillable="true"/>
<xs:complexType name="Products_information">
  <xs:complexContent>
    <xs:extension base="ex:entity">
      <xs:sequence>
        <xs:element name="productRecord">
          <xs:complexType>
            <xs:sequence>
              <xs:element ref="Product" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute ref="ex:itemType" fixed="Product"/>
            <xs:attribute ref="ex:cType" fixed="list"/>
            <xs:attribute ref="ex:arraySize" use="optional"/>
          </xs:complexType>
        </xs:element>
        <xs:element name="quantity" type="xs:decimal"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:element name="Product" type="Product" substitutionGroup="ex:entity" nillable="true"/>
<xs:complexType name="Product">
  <xs:complexContent>
    <xs:extension base="ex:entity">
      <xs:sequence>
        <xs:element name="id" type="xs:normalizedString"/>
        <xs:element name="name" type="xs:normalizedString" nillable="true" minOccurs="0"/>
        <xs:element name="description" type="xs:normalizedString" nillable="true" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

Figura 5.3 – XSD das entidades *Products\_information* e *Product*.

Deste modo, gerando as regras em Schematron, o resultado que é obtido está exemplificado na figura seguinte, onde são ilustradas as regras presentes na entidade do exemplo. A primeira regra "WR1" permite verificar se não existem elementos repetidos dentro da lista de produtos e a segunda regra permite verificar se a quantidade de produtos pedida é superior a zero.

```

<pattern name="productRecord - WR1">
  <rule context="Products_information/productRecord/*">
    <assert test="count(//Products_information[@id= current()/../@id]/productRecord/*[@ref =
      current()/@ref])=1">ERRO: VALUE_UNIQUE(productRecord)</assert>
  </rule>
</pattern>
<pattern name="Product_Record_Information - WR2">
  <rule context="quantity">
    <assert test="current() &gt; 0 or @ref">ERRO: quantity > 0 SELF</assert>
  </rule>
</pattern>

```

Figura 5.4 – Regras descritas em Schematron.

Com estas transformações efectuadas, é possível passar para o segundo passo, a geração de um serviço que permita efectuar testes de conformidade para as entidades descritas. Assim, com este serviço gerado, cada vez que o retalhista pretenda enviar um pedido de orçamento, contendo a listas de produtos e quantidade que pretendida, invoca o serviço de CT para verificar se o pedido está em conformidade com modelo definido.

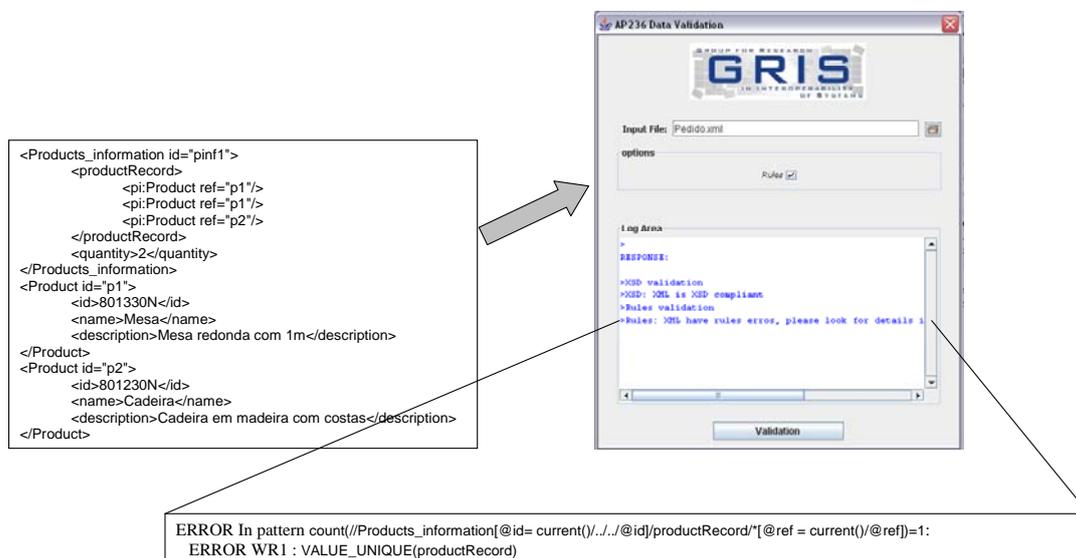


Figura 5.5 – Exemplo erro detectado pelo CT.

Se por acaso o ficheiro contiver um erro, como o exemplificado na figura anterior, onde o retalhista incluiu, por engano, dois elementos iguais na lista (dois "p1"), ao invocar o serviço

de CT, este envia um relatório com o erro encontrado, permitindo ao retalhista corrigir o erro antes de enviar o pedido para o fabricante.

Com o serviço de CT, tanto o retalhista como o fabricante, antes de enviarem uma mensagem ou quando receberem uma mensagem, podem invocar o serviço com o ficheiro de dados, recebendo o relatório com os erros encontrados, permitindo assim a correcção dos erros e reduzir os problemas na troca de dados.

## ***5.2. Cenário de aplicação, distribuição e entrega de produtos***

No projecto ATHENA, nomeadamente no sub-projecto A8, cujo propósito principal é a verificação da aplicabilidade das ferramentas desenvolvidas no projecto ATHENA em PMEs, foi escolhido um cenário de distribuição e entrega de produtos para caso de estudo [57].

Neste cenário, foram identificadas as principais barreiras de interoperabilidade que as PME sentem aos vários níveis: negócio, processos, serviços e troca de dados. Para cada uma das barreiras foram identificadas soluções conceptuais e tecnológicas que permitam superar a barreira.

O cenário proposto no projecto descreve os passos que uma PME de distribuição segue para fazer uma entrega de um produto. Neste processo, a empresa interage com empresas de transporte, normalmente de grandes dimensões, como a Fedex ou UPS.

O processo de distribuição e entrega, ilustrado na figura seguinte, começa pela ordem de venda, onde é descrito o produto a entregar, o endereço da venda, etc. Da ordem de venda é criada uma entrega, que pode ser agrupada com outras ordens caso seja possível a integração na rota desta entrega. Após definida a entrega é feita a recolha do produto e embalagem do mesmo, ficando assim pronto para seguir viagem.

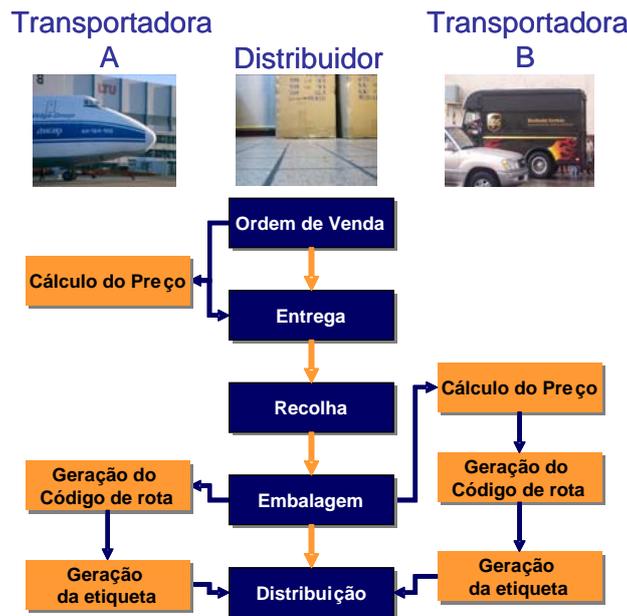


Figura 5.6 – Cenário 2 – Distribuição e entrega de produtos.

Em paralelo com estes processos, é necessário escolher qual a transportadora a utilizar para fazer a entrega. Como é possível ver na figura, onde são apresentadas duas transportadoras possíveis para fazer a distribuição, o processo de cálculo do preço e rota para entrega do produto, pode não ser feito da mesma maneira, nem na mesma altura. Deste modo, na transportadora A o cálculo do preço é feito logo a partir da ordem de venda, enquanto na B o cálculo do preço só é feito depois da embalagem do produto.

Ambas as empresas depois de terem a embalagem calculam a rota mais adequada. Esta rota entra em conta com o facto de poder haver mais de uma entrega, isto é, mais de um destino. Assim, a rota é feita de modo a efectuar um caminho mais adequado de distribuição. Depois da rota calculada é gerada a etiqueta identificativa do produto a entregar.

A empresa distribuidora, ao poder optar por várias empresas de transporte, necessita de escolher qual a mais adequada para cada entrega, permitindo baixar os custos e obter um maior rendimento. É nesta comunicação com outras empresas que as barreiras de interoperabilidade podem ocorrer.

Focando no caso particular dos problemas de interoperabilidade na troca de dados, foram identificados vários tipos de barreiras: a instanciação errada de modelos e diferentes restrições aos modelos [58].

A primeira barreira, a instanciação errada de modelos, pode ocorrer de várias formas, quer por erro humano na introdução dos dados ou por desconhecimento do modelo, quer por um erro de transmissão. Este tipo de erros podem provocar atrasos no processo ou mesmo entregas mal efectuadas.

A segunda barreira, encontra-se quando duas empresas que usam o mesmo modelo, mas implementam certo tipo de restrições nesse modelo, como por exemplo, para uma empresa a unidade de medida ser em polegadas para outra em metros ou uma precisar do peso do produto e a outra não. Este tipo de restrições faz com que quando é feita a troca de dados entre empresas, possam existir problemas, visto provocar a má interpretação dos dados ou falta deles.

Com vista a superar estas barreiras, foram identificadas quais as soluções que permitiam resolver estes problemas. Assim, para as barreiras descritas, constatou-se que a aplicação de testes de conformidade poderia ajudar a superar as barreiras, visto poder-se, através dos CT, assegurar a não existência de erros na troca de dados [59].

Deste modo a metodologia de testes de conformidade proposta nesta dissertação foi aplicada nas duas barreiras, embora de maneira diferentes e usando diferentes partes. Para ambos os casos foi utilizada a plataforma proposta nesta dissertação, ficando assim disponíveis serviços online para testes de conformidade.

Na primeira barreira, a parte da metodologia usada para resolver o problema, foi apenas a de teste dos dados contra o modelo definido em XSD e Schematron, usando a plataforma apresentada para efectuar estes testes e reportar os erros existentes a nível de instanciação

errada do modelo. Com a disponibilização destes serviços, uma empresa antes de enviar os dados para outra, invoca os serviços de testes de conformidade, recebendo o relatório com a existência ou não de erros.

Em relação à segunda barreira, muitas empresas desconhecem a possibilidade de representar restrições para a tecnologia XML, sendo estas restrições feitas normalmente “à mão”. Mas, como foi descrito na dissertação, é possível a inferência destas restrições através da linguagem Schematron, podendo-se assim verificar se os dados estão de acordo ou não com as restrições impostas.

A descrição das restrições em Schematron, permite que antes de enviar ou ao receber dados, as empresas possam efectuar testes de conformidade, permitindo verificar se as restrições impostas ao modelo estão a ser cumpridas. Como a definição destas restrições ou regras em Schematron pode não ser trivial, foi proposta também para ajudar a superar a barreira, para além da utilização dos testes de conformidade, a adopção da linguagem EXPRESS para descrever os modelos. A utilização do EXPRESS permite ao mesmo tempo descrever os modelos sintacticamente e semanticamente, podendo-se assim definir facilmente as restrições e a estrutura dos dados.

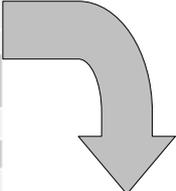
A definição do modelo em EXPRESS permite para além da definição completa do modelo, a interligação deste com normas STEP já existentes. Isto permite que as empresas possam utilizar AP já existentes para certificar os seus dados.

Como foi descrito, a plataforma apresentada nesta dissertação permite a transformação do modelo descrito em EXPRESS para XSD e Schematron, podendo assim utilizar os serviços para testes de conformidade.

No projecto A8, para verificar a aplicação destas soluções foi implementado um exemplo representativo do cenário: o pedido de orçamento para uma entrega por uma empresa distribuidora para uma empresa de transporte. Para o exemplo, foi desenvolvido o modelo

em EXPRESS, utilizando como base a descrição disponibilizada pela FEDEX (empresa transportadora) para este tipo de pedidos, como é possível ver na figura seguinte.

Element Name	Applicable Carrier	Valid For	Data Type	Max Length	Element Description/Requirements
<b>DECLAREDVALUE</b>					
DeclaredValue / Value	FDXE/FDXG	ALL	xs:decimal	8.2	Required for all shipments. The declared value of package contents. Format: Two explicit decimal positions (e.g. 100.00) 11 total length including decimal place.
DeclaredValue / CurrencyCode	FDXE/FDXG	ALL	xs:string	3	Optional. If Declared Value is provided, this element is required and must equal USD. Currency type used for customs value and declared/carriage value elements.
<b>REQUESTHEADER</b>					
RequestHeader / CustomerTransactionIdentifier	FDXE/FDXG	ALL	xs:string	40	Optional. This element allows a FedEx customer to assign a unique identifier to a transaction. It is returned in the Reply transaction to aid in matching transaction Requests to Replies.
RequestHeader / AccountNumber	FDXE/FDXG	ALL	xs:integer	12	The FedEx account number of the Sender is required to successfully receive a rate quote.
RequestHeader / MeterNumber	FDXE/FDXG	ALL	xs:integer	10	The unique meter number assigned to the Sender's FedEx account number is required to successfully receive a rate quote.
RequestHeader / CarrierCode	FDXE/FDXG	ALL	xs:string	4	Required. Valid values: FDXE - FedEx Express; FDXG - FedEx Ground



```

ENTITY DeclaredValue:
  Value : OPTIONAL NUMBER;
  CurrencyCode : OPTIONAL STRING;
  WHERE
  WR1: (NOT(EXISTS(Value)) OR (EXISTS(CurrencyCode) AND (CurrencyCode = 'USD')));
END_ENTITY;

ENTITY RequestHeader:
  CustomerTransactionIdentifier : OPTIONAL STRING;
  AccountNumber : INTEGER;
  MeterNumber : INTEGER;
  CarrierCode : CarrierCodeType;
  UNIQUE
  UR1 : CustomerTransactionIdentifier;
  UR2 : MeterNumber;
END_ENTITY;

TYPE CarrierCodeType = ENUMERATION OF
(FDXE,
 FDXG);
END_TYPE;

```

Figura 5.7 – Exemplo da elaboração do modelo STEP através da descrição da FEDEX.

Nesta figura é possível ver do lado superior esquerdo um excerto da descrição disponibilizada pela FEDEX, onde são descritas duas entidades, a “DECLAREDVALUE” e a “REQUESTHEADER”. A primeira entidade diz respeito ao preço da entrega, contendo dois atributos, um com o valor numérico e outro com a moeda. Na descrição dos atributos encontra-se uma restrição no uso destes atributos, que diz que se existir o atributo “Value” o atributo “CurrencyCode” deve conter “USD”, isto é, se for declarado um valor, a moeda usada deve ser o dólar americano.

A segunda entidade, “REQUESTHEADER”, é o cabeçalho do pedido de orçamento e contém os atributos: “CustomerTransactionIdentifier”, “AccountNumber”, “MeterNumber” e “CarrierNumber”. Na descrição destes atributos é possível encontrar duas restrições, uma explicitando que o atributo “CustomerTransactionIdentifier” tem de ser único e outra que o atributo “MeterNumber” tem igualmente que ser único.

Na parte inferior direita da figura anterior, é ilustrado um excerto do modelo EXPRESS construído através das descrições contidas no documento fornecido pela FEDEX. Nesta parte exemplificativa do modelo estão incluídas as duas entidades e as suas regras, descritas no exemplo. Como se pode observar foi necessário também, tendo em conta as descrições, criar um tipo “CarrierCodeType” que contém duas enumerações, “FDXE” e “FDXG”, usadas para identificar o código da entrega.

Após a definição do modelo este foi, utilizando a ferramenta proposta nesta dissertação, transformado em XSD e Schematron<sup>25</sup>. Na figura seguinte, continuando com o exemplo dado, está ilustrado o XSD e Schematron obtidos do modelo EXPRESS. Como é possível observar, foram criadas as duas entidades e um tipo no esquema XSD e no Schematron as três regras descritas.

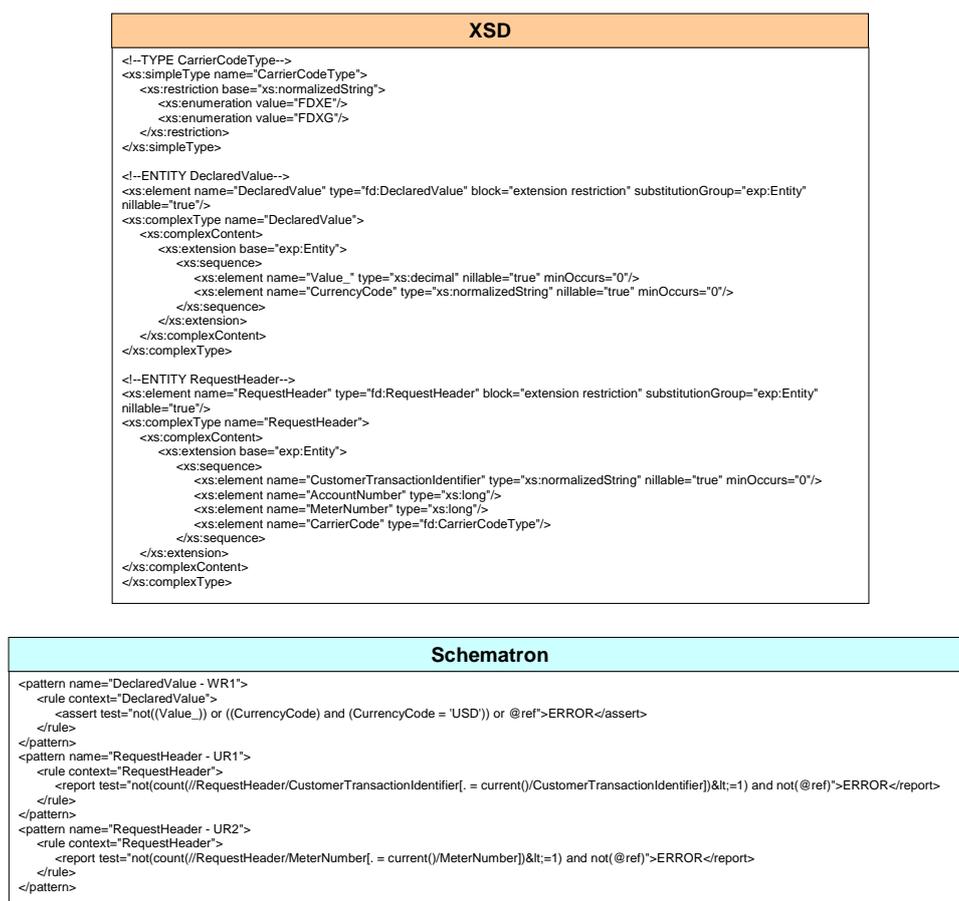


Figura 5.8 – XSD e Schematron obtidos a partir do exemplo.

<sup>25</sup> Tanto o modelo em STEP como os modelos XSD e Schematron obtido podem ser consultados no Anexo 2

Estes dois modelos foram utilizados para formar a base de conhecimento que permite efectuar os testes de conformidade sobre os dados trocados. O CT ficou disponível através de um web-service, tal como descrito na plataforma, permitindo assim o acesso ao serviço através da Internet por parte da aplicação.

Acedendo ao serviço, a empresa de distribuição pode verificar a existência de erros no seu pedido em XML. Com o relatório recebido após a invocação do serviço de CT, a empresa, pode corrigir os erros antes de enviar o pedido para a outra empresa. Na figura seguinte é ilustrada uma aplicação desenvolvida para simular a troca de dados entre a empresa de distribuição e a transportadora. Na figura é possível observar que a empresa de distribuição invocou o serviço, tendo obtido como resposta que o ficheiro que pretendia enviar continha erros. Ao receber o relatório com a descrição dos erros, a distribuidora pode corrigi-los e após verificar novamente que os dados estão certos, enviá-los para a transportadora.

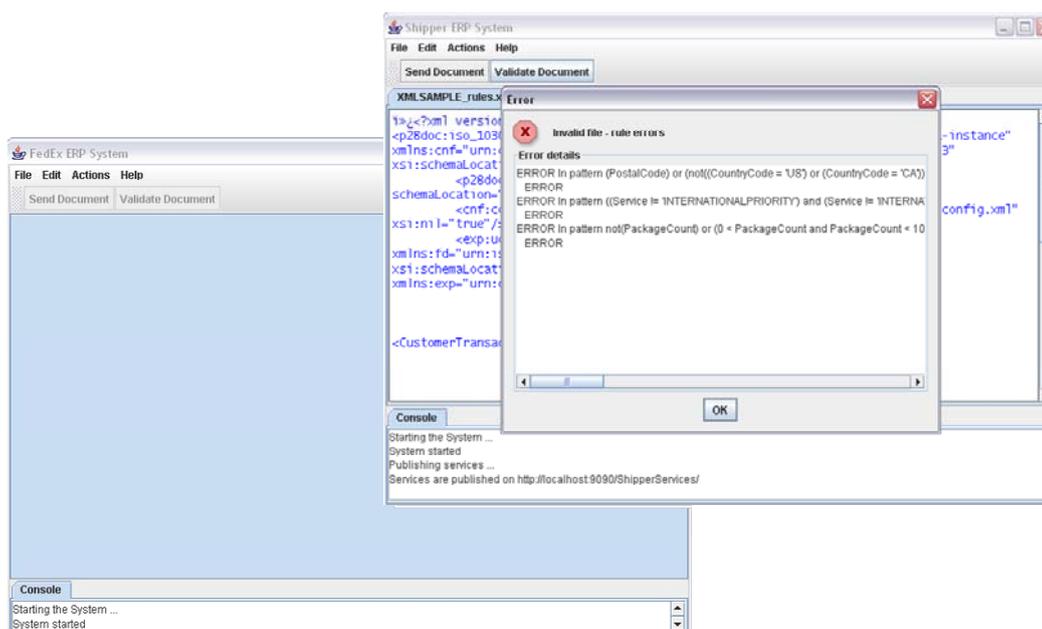


Figura 5.9 – Exemplo de troca de dados entre as empresas

O desenvolvimento destas aplicações permitiu simular uma troca de mensagens entre empresas, verificando as vantagens do uso dos testes de conformidade para comprovar que os dados trocados entre as duas empresas cumprem todas as especificações descritas no modelo EXPRESS, quer a nível sintáctico quer a nível semântico.

## **Conclusões e Perspectivas de Trabalho Futuro**

A evolução tecnológica não deixa ninguém indiferente e nos últimos anos esta tem vindo a acentuar-se no mundo informático. Neste contexto, a Internet tem vindo a ocupar o centro das atracções, podendo-se hoje em dia fazer quase tudo através da Internet, desde movimentar a conta bancária até fazer as compras do supermercado.

Com as portas abertas para um mundo novo de oportunidades, as empresas têm vindo a tentar aproveitar estes avanços informáticos para prosperar nos seus negócios. Estando esta evolução acompanhada da liberalização do comércio mundial, as oportunidades de negócio, o estabelecimento de novas parcerias e aumento dos benefícios, tornam-se ainda maiores.

Assim, sobretudo as PMEs, mais sujeitas às alterações no mundo de negócios devido ao seu tamanho, vêm-se obrigadas a seguir as novas tendências do mercado e a adoptar as novas tecnologias. Nesta procura para acompanhar o processo de globalização, as PME têm vindo a estabelecer novas parcerias e a procurar novas formas de valorizar os seus produtos e serviços.

Na interacção com as outras empresas, quando se estabelecem novas parcerias, as empresas podem encontrar vários problemas, como o problema da falta de interoperabilidade entre os seus sistemas informáticos e com os das outras empresas. Uma das formas de minimizar este problema é a adopção de normas para a troca de dados.

No que cabe à troca e a integração de dados de produto, a comunidade ISO, tem vindo a desenvolver o conjunto de normas STEP com vista a ajudar a resolver os problemas de interoperabilidade dos sistemas que pretendam trocar dados de produto e ao mesmo tempo certificar a correcção desses dados com as normas desenvolvidas.

No entanto, embora bastante difundido nas empresas, existem poucas ferramentas para troca e manipulação de dados STEP. Em contrapartida, formatos como o XML, estão agora

em expansão da sua implementação e utilização, sendo esta uma linguagem de fácil utilização e existindo bastantes aplicações sem custos.

O XML, ao ser um dos formatos mais usados para representação e troca de dados na Internet, tem vindo a ser adoptado por parte das empresas nas suas trocas de dados. No entanto, existem empresas que pretendem usar ao mesmo tempo o STEP e XML, visto que o STEP permite certificar os dados de produto pela ISO e facilitar a interoperabilidade dos seus sistemas e o XML permite baixar os custos de implementações, visto existir muitas implementações disponíveis para este formato, e ser um formato fácil de entender e usar.

A ISO desenvolveu a Parte 28, usada como referência para mapear os modelos de STEP para XSD, permitindo assim usar o XML para instanciar dados. No entanto, nem toda a informação que consta nestes modelos é transferida para o XSD, visto este não suportar regras, restrições, funções e procedimentos.

Esta dissertação, tendo em vista ajudar a adopção de normas STEP e ao mesmo tempo ser possível usar XML para representar dados, desenvolveu uma ferramenta que permite transformar a informação presente nos modelos em EXPRESS, que não pode ser transformada XSD, para uma tecnologia capaz de complementar o XSD (Schematron), contribuindo assim para a interoperabilidade e conformidade de sistemas STEP com dados XML.

Com os modelos transformados por completo para uma XSD, através da Parte 28 e contendo a parte sintáctica dos modelos, e para Schematron, contendo a parte semântica dos modelos, é então possível validar por completo os dados XML contra os modelos conceptuais STEP.

Com vista a ajudar na implementação de testes de conformidade e verificação de interoperabilidade, por parte das empresas que adoptem normas STEP e representem os

dados em XML, nesta dissertação é proposta uma extensão da metodologia descrita nas Partes 30s do STEP permitindo realizar testes de conformidade e verificação de interoperabilidade a estes sistemas.

Com base nesta metodologia foi desenvolvida uma plataforma SOA para implementação de testes de conformidade e verificação de interoperabilidade, ficando estas validações disponíveis na Internet através de serviços. A disponibilização destes serviços, permite entre outras coisas, a sua fácil integração em sistemas já existentes nas empresas e por outro lado, ficar disponível a qualquer empresa desde que ligada à Internet.

A plataforma desenvolvida foi testada e validada em dois cenários industriais, tendo-se verificado a enorme ajuda que esta pode dar às empresas no estabelecimento de novas parcerias e na certificação da conformidade dos seus dados, ajudando assim as empresas a não perder o comboio da globalização.

A ferramenta de transformação dos modelos em EXPRESS para XSD e Schematron é um protótipo que foi desenvolvido de acordo com limitação de tempo disponível para o desenvolvimento deste trabalho de mestrado. O trabalho futuro considera terminar esta ferramenta, podendo assim integrá-la por completo na plataforma SOA de testes de conformidade para a norma ISO 10303-AP236. Existe também trabalho a desenvolver na plataforma de CT, faltando integrar por completo os ATC desenvolvidos em TTCN-3.

Um dos desafios que está também considerado para melhoramento do trabalho desenvolvido, considera uma evolução na arquitectura do sistema apresentada de forma a permitir a elaboração de novos relatórios que permitam processamento e correcção automatizada dos erros encontrados nos dados XML.

O serviço de CT também pode sofrer melhoramentos, tais como a possibilidade de incluir qualquer modelo EXPRESS em anexo e executar os testes de conformidade contra esse modelo, não ficando assim limitado à base de dados existente.

A disponibilização online da ferramenta de transformação, permitindo assim que qualquer utilizador possa transformar o seu modelo EXPRESS em XSD e Schematron, pode ser outra das implementações futuras que pode dar um valor acrescido a esta dissertação.

Por fim, a elaboração de uma base de dados com todos os AP transformados para os seus equivalentes em XSD e Schematron, poderá ajudar as empresas a adoptarem cada vez mais estas normas ISO.

## Bibliografia

- [1] Jardim-Goncalves R., Maló P., Vieira H., Steiger-Garção A., "Improving competitiveness through SMART Furniture Manufacturing in Extended Environments", *Advanced Design, Production and Management Systems: ISBN 90 5809 524 6; 10th ISPE International Conference On Concurrent Engineering: Research And Applications*; 2003
- [2] Institute of Electrical and Electronics Engineers. IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. New York, NY: 1990.
- [3] Consórcio Athena (2007), "Guidelines and Best Practices for Applying the ATHENA Interoperability Framework to Support SME Participation in Digital Ecosystems", *Deliverable A8.2*.
- [4] Comissão Europeia (2003), Linking up Europe: The Importance of Interoperability for eGovernment Services
- [5] Comissão Europeia (2000), "Estratégia de Lisboa", <http://ec.europa.eu/growthandjobs/>
- [6] Comissão Europeia (2006), "Online Availability of Public Services: How Is Europe Progressing?"
- [7] Mark Roberti, "The Importance of Standardization", RFID Journal, 2004
- [8] International Standards Organization (ISO), "Overview of the ISO system", <http://www.iso.org/iso/en/aboutiso/introduction/index.html>, 2007
- [9] PDES, Inc. Consortium Waives Membership Fees for Small Vendors and Users; [www.aticorp.org/pres\\_pdes\\_join3.htm](http://www.aticorp.org/pres_pdes_join3.htm); 2004
- [10] M. P. Gallaher, A. C. O'Connor, T. Phelps (Dec. 2002), Economic Impact Assessment of International Standard for the Exchange of Product Model Data (STEP) in Transportation Equipment Industries, NIST Planning Report 02-5, [http://www.mel.nist.gov/msid/sima/step\\_economic\\_impact.pdf](http://www.mel.nist.gov/msid/sima/step_economic_impact.pdf)
- [11] ISO 10303-21:2002, Industrial automation systems and integration—Product data representation and exchange—Part 21: Implementation methods: Clear text encoding of the exchange structure
- [12] Lubell, J.; Peak, R. S.; Srinivasan, V; Waterbury S. C.; "STEP, XML, AND UML: COMPLEMENTARY TECHNOLOGIES"; ASME 2004 Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC 2004); Salt Lake City, Utah USA, Sep 28- Oct 2, 2004
- [13] W3C Consortium: "Extensible Markup Language (XML)"; [www.w3.org/XML](http://www.w3.org/XML), 2007
- [14] Artur Afonso de Sousa, "Bases de Dados, Web e XML", FCA – Editora de informática, 2002

- [15] ISO10303-11:2002, 'Product data representation and exchange—Part 11: Description methods: The EXPRESS language reference manual'
- [16] W3C Consortium: "XML Schema Definition (XSD)"; [www.w3.org/XML/Schema](http://www.w3.org/XML/Schema), 2007
- [17] Schematron - <http://www.schematron.com/>, 2007
- [18] Wikipédia, [http://pt.wikipedia.org/wiki/Globalização](http://pt.wikipedia.org/wiki/Globaliza%C3%A7%C3%A3o), 2007
- [19] Martin Wolf, "Why Globalization Works", 2005
- [20] Comissão Europeia (2005), "i2010 - A European Information Society for growth and employment", [http://ec.europa.eu/information\\_society/eeurope/i2010/index\\_en.htm](http://ec.europa.eu/information_society/eeurope/i2010/index_en.htm)
- [21] Comissão Europeia (2002), "eEurope 2005 – An Information society for all", [http://ec.europa.eu/information\\_society/eeurope/2005/index\\_en.htm](http://ec.europa.eu/information_society/eeurope/2005/index_en.htm)
- [22] Ray, S.; "Interoperability Standards in the Semantic Web"; Journal of Computer and Information Science in Engineering, vol. 2; March, 2002
- [23] International Organization for Standardization (ISO), <http://www.iso.org>, 2007
- [24] Canada's Business and Consumer site - <http://strategis.ic.gc.ca/engdoc/main.html>, 2007
- [25] Plant Information Interchange via STEP - <http://cic.nist.gov/plantstep/index.html>, 2007
- [26] Jardim-Gonçalves, R.; "A Framework for multilevel Standard protocols and Interoperability"; Tese de doutoramento, 2004
- [27] Gilbert Arbez, Leigh Swain; "OSI Conformance Testing for Bibliographic applications"
- [28] Consórcio SMART-fm (2003), 'Annex 1 – Description of Work'; <http://www.smart-fm.funstep.org>
- [29] funStep Interest Group (2006), <http://www.funstep.org>
- [30] ISO10303-236 (2007), 'STEP Part 236, Application Protocol: Furniture product data and project data'
- [31] Consórcio Athena (2004), 'Annex I - Description of Work', [www.athena-ip.org](http://www.athena-ip.org)
- [32] STEP on a Page - <http://www.mel.nist.gov/sc5/soap/>, 2007
- [33] Howard M. Bloom, "What Is STEP and How Is It Used?", 1992
- [34] Jardim-Gonçalves, R., Agostinho, C., Malo, P., and Steiger-Garção, A.; "Harmonizing Technologies in Conceptual Models Representation"
- [35] Jardim-Gonçalves, R., Agostinho, C., Malo, P., and Steiger-Garção, A.; "AP236-XML: A Framework for integration and harmonization of STEP Application Protocols", ASME 2005

- [36] ISO TC184/SC4/WG11 N223 (2004), 'Product data representation and exchange: Implementation methods: XML Schema governed representation of EXPRESS schema governed data', 2004
- [37] <http://www.idealliance.org/proceedings/xml04/papers/81/xml-2004-hu.html>, 2004
- [38] <http://www.di.uminho.pt/~jcr/PROJS/xcsi-www/>, 2007
- [39] Khaue Rezende Rodrigues, Ronaldo dos Santos Mello; "A Faceted Taxonomy of Semantic Integrity Constraints for the XML Data Model", DEXA 2007
- [40] Jardim-Gonçalves, R., Onofre, S., and Agostinho, C. (2006), "Conformance Testing for XML-based STEP Conceptual Models", ASME 2006 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Philadelphia, Setembro, 2006
- [41] ISO/IEC 7498-1(1994), "Information technology - Open Systems Interconnection - Basic Reference Model: The Basic Model"
- [42] ISO/IEC Standard 9646:1991, Open Systems Conformity Testing Methodology and Framework
- [43] Jan Tretmans Formal Methods & Tools group, University of Twente, "An Overview of OSI Conformance Testing", 2001
- [44] International Engineering Consortium; Tree and Tabular Combined Notation—The Language, 2007
- [45] <http://pt.wikipedia.org>, 2007
- [46] U.S. Department of Defense Biometrics Management Office, "BioAPI Conformance Test Suite", 2006
- [47] Alexandru Czimbor, "Biometric Standards & Standards Based Tools", CAIP 2004
- [48] ISO 10303 Standard for Exchange of Product Data (STEP), ISO TC184/SC4, Part31, Conformance testing methodology and framework: General concepts, International Organization for Standardization, 2004
- [49] Anthony Wiles, Relevance of Conformance Testing for Interoperability Testing, ACATS ATS-CONF 2003
- [50] ETSI, Making better Standards, <http://portal.etsi.org/mbs/introduction/Where.asp>, 2007
- [51] Scott Moseley; Steve Randall; Anthony Wiles, In Pursuit of Interoperability, J. of IT Standards & Standardization Research, 2004
- [52] TTCN-3, <http://www.ttcn-3.org/>, 2007
- [53] Scott Moseley; Steve Randall; Anthony Wiles, "Experience Within ETSI of the Combined Roles of Conformance Testing and Interoperability Testing", IEEE 2003

- [54] S. Onofre, C. Agostinho, R. Jardim-Gonçalves, and A. Steiger-Garção, "Applying TTCN to enhance B2B Conformance testing frameworks", I-ESA 2007 3rd International Conference on Interoperability for Enterprise Software and Applications, 2007
- [55] ISO 10303 Standard for Exchange of Product Data (STEP), ISO TC184/SC4, Part34, Conformance testing methodology and framework: Conformance testing methodology and framework: Abstract test methods for SDAI implementations, International Organization for Standardization, 2003
- [56] Consórcio Athena (2004), 'Piloting including Technology Testing Coordination and Pilot Infrastructure', *Deliverable B5.1*.
- [57] Consórcio Athena (2006), 'Enterprise Modelling in the Context of Collaborative Enterprises', *Working Document WD.A8.1*.
- [58] Consórcio Athena (2007), 'Guidelines and Best Practices for Applying the ATHENA Interoperability Framework to Support SME Participation in Digital Ecosystems', *Deliverable D.A8.2*.
- [59] Consórcio Athena (2007), 'Updated ATHENA tools and infrastructures supporting B5 Sub Projects', *Deliverable D.A8.3*.



<b>attribute_qualifier</b> = '.' attribute_ref	<b>attribute_qualifier</b> = '.' attribute_ref
<b>group_qualifier</b> = '\' entity_ref	<b>group_qualifier</b> = '\' entity_ref
<b>index_qualifier</b> = '[' index_1 [ ':' index_2 ] ]'	<b>index_qualifier</b> = '[' index_1 [ ':' index_2 ] ]'
<b>index</b> = numeric_expression	<b>index</b> = numeric_expression
<b>add_like_op</b> = '+'   '-'   OR   XOR	<b>add_like_op</b> = '+'   '-'   'or'   'xor'
<b>built_in_constant</b> = CONST_E   PI   SELF   '?'	<b>built_in_constant</b> = CONST_E   PI   current()   '?'
<b>built_in_function</b> = ABS   ACOS   ASIN   ATAN   BLENGTH   COS   EXISTS   EXP   FORMAT   HIBOUND   HIINDEX   LENGTH   LOBOUND   LOINDEX   LOG   LOG2   LOG10   NVL   ODD   ROLESOF   SIN   SIZEOF   SQRT   TAN   TYPEOF   USEDIN   VALUE   VALUE_IN   VALUE_UNIQUE	<b>built_in_function</b> = 'math:abs'   'math:acos'   'math:asin'   'math:atan'   BLENGTH   'math:cos'   ''   'math:exp'   FORMAT   'math:max'   'math:highest'   LENGTH   'math:min'   'math:lowest'   'math:log'   LOG2   LOG10   NVL   ODD   ROLESOF   'math:sin'   'count'   'math:sqrt'   'math:tan'   TYPEOF   USEDIN   VALUE   VALUE_IN   VALUE_UNIQUE
<b>interval</b> = '{' interval_low interval_op interval_item interval_op interval_high '}'	<b>interval</b> = interval_low interval_op interval_item ' and ' interval_item interval_op interval_high
<b>interval_high</b> = simple_expression	<b>interval_high</b> = simple_expression
<b>interval_item</b> = simple_expression	<b>interval_item</b> = simple_expression
<b>interval_low</b> = simple_expression	<b>interval_low</b> = simple_expression
<b>interval_op</b> = '<'   '<='	<b>interval_op</b> = '&lt;'   '&lt;='
<b>multiplication_like_op</b> = '*'   '/'   DIV   MOD   AND   '  '	<b>multiplication_like_op</b> = '*'   '/'   DIV   MOD   and   '  '
<b>rel_op</b> = '<'   '>'   '<='   '>='   '<>'   '='   '<>:'   ':='	<b>rel_op</b> = '&lt;'   '&gt;'   '&lt;='   '&gt;='   '!= '   '= '   '!= '   '= '
<b>rel_op_extended</b> = rel_op   IN   LIKE	<b>rel_op_extended</b> = rel_op   '= '   LIKE
<b>unary_op</b> = '+'   '-'   NOT	<b>unary_op</b> = '+'   '-'   'not'

## Anexo 2 – Exemplo de transformação de um modelo STEP

### Exemplo de EXPRESS

SCHEMA FDXRate;

TYPE CarrierCodeType = ENUMERATION OF  
(FDXE,  
FDXG);  
END\_TYPE;

TYPE CODType = ENUMERATION OF  
(ANY,  
GUARANTEEDFUNDS,  
CASH);  
END\_TYPE;

TYPE DangerousGoodsStatus = ENUMERATION OF  
(ACCESSIBLE,  
INACCESSIBLE);  
END\_TYPE;

TYPE Dropoff = ENUMERATION OF  
(REGULARPICKUP,  
REQUESTCOURIER,  
DROPBOX,  
BUSINESSSERVICE\_CENTER,  
STATION);  
END\_TYPE;

TYPE HomeDeliveryType = ENUMERATION OF  
(DATECERTAIN,  
EVENING,  
APPOINTMENT);  
END\_TYPE;

TYPE measureUnit = EXTENSIBLE ENUMERATION;  
END\_TYPE;

TYPE LenghtUnit = ENUMERATION BASED\_ON measureUnit WITH  
(IN\_  
CM);  
END\_TYPE;

TYPE Package = ENUMERATION OF  
(FEDEXENVELOPE,  
FEDEXPAK,  
FEDEXBOX,  
FEDEXTUBE,  
FEDEX10KGBOX,  
FEDEX25KGBOX,  
YOURPACKAGING);  
END\_TYPE;

TYPE Services = ENUMERATION OF  
(PRIORITYOVERNIGHT,  
STANDARDOVERNIGHT,  
FIRSTOVERNIGHT,

```
FEDEX2DAY,  
FEDEXEXPRESSSAVER,  
INTERNATIONALPRIORITY,  
INTERNATIONALECONOMY,  
INTERNATIONALFIRST,  
FEDEX2DAYFREIGHT,  
FEDEX3DAYFREIGHT,  
FEDEXGROUND,  
GROUNDHOMEDELIVERY,  
INTERNATIONALPRIORITY_FREIGHT,  
INTERNATIONALECONOMY_FREIGHT,  
EUROPEFIRSTINTERNATIONALPRIORITY);  
END_TYPE;
```

```
TYPE ShipmentIndicator = ENUMERATION OF  
(NONRETURN,  
PRINTRETURNLABEL,  
EMAILLABEL,  
FEDEXTAG);  
END_TYPE;
```

```
TYPE SignatureType = ENUMERATION OF  
(DELIVERWITHOUTSIGNATURE,  
INDIRECT,  
DIRECT,  
ADULT);  
END_TYPE;
```

```
TYPE VariableHandlingChargesLevel = ENUMERATION OF  
(PACKAGE_  
SHIPMENT);  
END_TYPE;
```

```
TYPE VariableHandlingChargesType = ENUMERATION OF  
(FIXED_AMOUNT,  
PERCENTAGE_OF_BASE,  
PERCENTAGE_OF_NET,  
PERCENTAGE_OF_NET_EXCL_TAXES);  
END_TYPE;
```

```
TYPE WeightUnit = ENUMERATION BASED_ON measureUnit WITH  
(LBS,  
KGS);  
END_TYPE;
```

```
ENTITY Address;  
  StateOrProvinceCode : OPTIONAL STRING;  
  PostalCode          : OPTIONAL STRING;  
  CountryCode         : STRING;  
  WHERE  
    WR1: (EXISTS(StateOrProvinceCode) OR NOT(((CountryCode = 'US') OR  
(CountryCode = 'CA'))));  
    WR2: (EXISTS(PostalCode) OR NOT(((CountryCode = 'US') OR (CountryCode =  
'CA'))));  
END_ENTITY;
```

```
ENTITY CalendarDate;  
  day_component : DayInMonthNumber;
```

```

    month_component : MonthInYearNumber;
    year_component : YearNumber;
WHERE
    WR1 : ValidCalendarDate (SELF);
END_ENTITY;

ENTITY COD;
    CollectionAmount : NUMBER;
    CollectionType : CODType;
END_ENTITY;

ENTITY DangerousGoods;
    Accessibility : DangerousGoodsStatus;
END_ENTITY;

ENTITY DeclaredValue;
    Value_ : OPTIONAL NUMBER;
    CurrencyCode : OPTIONAL STRING;
WHERE
    WR1: (NOT(EXISTS(Value_)) OR (EXISTS(CurrencyCode) AND (CurrencyCode =
'USD')));
END_ENTITY;

ENTITY DestinationAddress
    SUBTYPE OF (Address);
END_ENTITY;

ENTITY Dimensions;
    Length_ : OPTIONAL STRING;
    Width : OPTIONAL STRING;
    Height : OPTIONAL STRING;
    Units : LengthUnit;
WHERE
    WR1: (NOT(EXISTS(Units))) OR ((EXISTS(Length_) OR (EXISTS(Width)) OR
(EXISTS(Height)));
END_ENTITY;

ENTITY FDXRateRequest;
    RequestHeader : RequestHeader;
    ReturnShipmentIndicator : OPTIONAL ShipmentIndicator;
    ShipDate : CalendarDate;
    DropoffType : Dropoff;
    Service : Services;
    Packaging : Package;
    WeightUnits : WeightUnit;
    Weight : NUMBER;
    ListRate : OPTIONAL BOOLEAN;
    OriginAddress : OriginAddress;
    DestinationAddress : DestinationAddress;
    Payment : OPTIONAL Payment;
    Dimensions : OPTIONAL Dimensions;
    DeclaredValue : OPTIONAL DeclaredValue;
    Alcohol : OPTIONAL BOOLEAN;
    SpecialServices : OPTIONAL SpecialServices;
    HomeDelivery : OPTIONAL HomeDelivery;
    PackageCount : OPTIONAL INTEGER;
    VariableHandlingCharges : OPTIONAL VariableHandlingCharges;
    International : OPTIONAL International;
WHERE

```

```

WR1: (((Service <> Services.INTERNATIONALPRIORITY) AND (Service <>
Services.INTERNATIONALECONOMY) AND (Service <>
Services.INTERNATIONALFIRST) AND (Service <>
Services.INTERNATIONALPRIORITY_FREIGHT) AND (Service <>
Services.INTERNATIONALECONOMY_FREIGHT) AND (Service <>
Services.EUROPEFIRSTINTERNATIONALPRIORITY)) OR ((DropoffType =
Dropoff.REGULARPICKUP) OR (DropoffType =
Dropoff.REQUESTCOURIER) OR (DropoffType = Dropoff.STATION)));
WR2: NOT(EXISTS(PackageCount) or { 0<PackageCount <1000});
END_ENTITY;

```

```

ENTITY HomeDelivery;
  Type_ : HomeDeliveryType;
END_ENTITY;

```

```

ENTITY International;
  BrokerSelectionOption : BOOLEAN;
END_ENTITY;

```

```

ENTITY OriginAddress
  SUBTYPE OF (Address);
END_ENTITY;

```

```

ENTITY Payment;
  PayorType : STRING;
END_ENTITY;

```

```

ENTITY RequestHeader;
  CustomerTransactionIdentifier : OPTIONAL STRING;
  AccountNumber : INTEGER;
  MeterNumber : INTEGER;
  CarrierCode : CarrierCodeType;
  UNIQUE
  UR1 : CustomerTransactionIdentifier;
  UR2 : MeterNumber;
END_ENTITY;

```

```

ENTITY SpecialServices;
  COD : OPTIONAL COD;
  HoldAtLocation : BOOLEAN;
  DangerousGoods : DangerousGoods;
  DryIce : BOOLEAN;
  ResidentialDelivery : BOOLEAN;
  ResidentialPickup : OPTIONAL BOOLEAN;
  InsidePickup : BOOLEAN;
  InsideDelivery : BOOLEAN;
  SaturdayPickup : BOOLEAN;
  SaturdayDelivery : BOOLEAN;
  NonstandardContainer : BOOLEAN;
  SignatureOption : SignatureType;
END_ENTITY;

```

```

ENTITY VariableHandlingCharges;
  Level : OPTIONAL VariableHandlingChargesLevel;
  Type_ : OPTIONAL VariableHandlingChargesType;
  AmountOrPercentage : OPTIONAL NUMBER;
  WHERE
  WR1: (NOT(EXISTS(Type_)) OR (EXISTS(Level)));

```

```

WR2: (NOT(EXISTS(Type_)) OR (EXISTS(AmountOrPercentage)));
END_ENTITY;

FUNCTION LeapYear (Year : YearNumber): BOOLEAN;
  IF (((Year MOD 4) = 0) AND ((Year MOD 100) <> 0)) OR ((Year MOD 400) = 0)
  THEN RETURN(TRUE);
  ELSE
    RETURN(FALSE);
  END_IF;
END_FUNCTION;

FUNCTION ValidCalendarDate (Date : CalendarDate): LOGICAL;
  IF NOT ({1 <= Date.day_component <= 31}) THEN
    RETURN(FALSE);
  END_IF;
  CASE Date.month_component OF
    4      : RETURN({ 1<= Date.day_component <= 30});
    6      : RETURN({ 1<= Date.day_component <= 30});
    9      : RETURN({ 1<= Date.day_component <= 30});
    11     : RETURN({ 1<= Date.day_component <= 30});
    2      : BEGIN
      IF (LeapYear(Date.year_component)) THEN
        RETURN({ 1<= Date.day_component <= 29});
      ELSE
        RETURN({ 1<= Date.day_component <= 28});
      END_IF;
    END;
  OTHERWISE : RETURN(TRUE);
  END_CASE;
END_FUNCTION;

END_SCHEMA;

```

### XSD Obtido

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:oid:1.0.10303.28.2.1.1"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:exp="urn:oid:1.0.10303.28.2.1.1">
  <!-- ROOT ELEMENT UOS -->
  <xs:complexType name="uos">
    <xs:sequence>
      <xs:element ref="exp:header" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:ID" use="optional"/>
    <xs:attribute name="express" type="exp:Seq-anyURI" use="optional"/>
    <xs:attribute name="configuration" type="exp:Seq-anyURI" use="optional"/>
    <xs:attribute name="schemaLocation" type="exp:Seq-anyURI"
use="optional"/>
    <xs:attribute name="edo" type="xs:anyURI" use="optional"/>
    <xs:attribute name="defaultLanguage" type="xs:language" use="optional"/>
  </xs:complexType>
  <xs:element name="uos" type="exp:uos">
    <xs:key name="base64Binary-key">
      <xs:selector xpath="exp:base64Binary-wrapper"/>
      <xs:field xpath="@id"/>
    </xs:key>
    <xs:key name="hexBinary-key">
      <xs:selector xpath="exp:hexBinary-wrapper"/>
      <xs:field xpath="@id"/>
    </xs:key>
  </xs:element>

```

```

</xs:key>
<xs:key name="logical-key">
  <xs:selector xpath="exp:logical-wrapper"/>
  <xs:field xpath="@id"/>
</xs:key>
<xs:key name="integer-key">
  <xs:selector xpath="exp:integer-wrapper"/>
  <xs:field xpath="@id"/>
</xs:key>
<xs:key name="boolean-key">
  <xs:selector xpath="exp:boolean-wrapper"/>
  <xs:field xpath="@id"/>
</xs:key>
<xs:key name="double-key">
  <xs:selector xpath="exp:double-wrapper"/>
  <xs:field xpath="@id"/>
</xs:key>
<xs:key name="decimal-key">
  <xs:selector xpath="exp:decimal-wrapper"/>
  <xs:field xpath="@id"/>
</xs:key>
<xs:key name="long-key">
  <xs:selector xpath="exp:long-wrapper"/>
  <xs:field xpath="@id"/>
</xs:key>
<xs:key name="string-key">
  <xs:selector xpath="exp:string-wrapper"/>
  <xs:field xpath="@id"/>
</xs:key>
</xs:element>
<!-- UOS and DOCUMENT HEADER -->
<xs:complexType name="name_and_address">
  <xs:sequence>
    <xs:element name="name" type="xs:string"/>
    <xs:element name="address">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="address_line" type="xs:string"
minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:element name="header">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="name" type="xs:string" minOccurs="0"/>
      <xs:element name="time_stamp" type="xs:dateTime"
minOccurs="0"/>
      <xs:element name="author" type="exp:name_and_address"
minOccurs="0"/>
      <xs:element name="organization"
type="exp:name_and_address" minOccurs="0"/>
      <xs:element name="preprocessor_version" type="xs:string"
minOccurs="0"/>
      <xs:element name="originating_system" type="xs:string"
minOccurs="0"/>

```

```

        <xs:element name="authorization" type="xs:string"
minOccurs="0"/>
        <xs:element name="documentation" type="xs:string"
minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:attribute name="authority" type="xs:anyURI"/>
<xs:complexType name="edokey" abstract="true">
    <xs:attribute name="id" type="xs:ID" use="required"/>
    <xs:attribute name="edo" type="xs:anyURI" use="optional"/>
    <xs:attribute ref="exp:authority" use="required"/>
</xs:complexType>
<xs:attribute name="edokeyType" type="xs:QName"/>
<xs:element name="edokey" type="exp:edokey" abstract="true"/>
<!-- EXP ENTITY DEFINITIONS and DECLARATIONS -->
<xs:complexType name="Entity" abstract="true">
    <xs:annotation>
        <xs:documentation>The Entity type is the supertype for every
            EXPRESS entity. It provides the exp id attribute that is the local
            identifier for each instance as well as the other exp descriptive attributes.
        </xs:documentation>
    </xs:annotation>
    <xs:attribute name="href" type="xs:anyURI" use="optional"/>
    <xs:attribute name="ref" type="xs:IDREF" use="optional"/>
    <xs:attribute name="proxy" type="xs:IDREF" use="optional"/>
    <xs:attribute name="edo" type="xs:anyURI" use="optional"/>
    <xs:attributeGroup ref="exp:instanceAttributes"/>
</xs:complexType>
<xs:element name="Entity" type="exp:Entity" abstract="true" nillable="true"/>
<xs:element name="complexEntity" substitutionGroup="exp:Entity">
    <xs:complexType>
        <xs:complexContent>
            <xs:extension base="exp:Entity">
                <xs:sequence>
                    <xs:element ref="exp:Entity" minOccurs="0"
maxOccurs="unbounded"/>
                    <xs:element ref="exp:Single-Entity"
minOccurs="0"
maxOccurs="unbounded"/>
                </xs:sequence>
                <xs:attribute name="entities" type="xs:NMTOKENS"
use="required"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
</xs:element>
<xs:element name="Single-Entity" type="exp:Single-Entity" abstract="true"/>
<xs:complexType name="Single-Entity"/>
<!-- UNRESTRICTED AGGREGATION TYPE DEFINITIONS -->
<xs:complexType name="Seq-IDREF">
    <xs:simpleContent>
        <xs:extension base="xs:IDREFS">
            <xs:attribute name="ref" type="xs:IDREF" use="optional"/>
            <xs:attribute ref="exp:itemType" use="optional"/>
            <xs:attribute ref="exp:cType" use="optional"/>
            <xs:attribute ref="exp:arraySize" use="optional"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>

```

```

<xs:complexType name="Seq-IDREF-wrapper">
  <xs:simpleContent>
    <xs:extension base="exp:Seq-IDREF">
      <xs:attributeGroup ref="exp:instanceAttributes"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<!-- GLOBAL ATTRIBUTE DECLARATIONS -->
<!-- EXP ARRAY -->
<xs:attribute name="arraySize">
  <xs:simpleType>
    <xs:restriction>
      <xs:simpleType>
        <xs:list itemType="xs:integer"/>
      </xs:simpleType>
      <xs:minLength value="1"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
<xs:attribute name="itemType">
  <xs:simpleType>
    <xs:list itemType="xs:QName"/>
  </xs:simpleType>
</xs:attribute>
<xs:attribute name="cType">
  <xs:simpleType>
    <xs:list itemType="exp:aggregateType"/>
  </xs:simpleType>
</xs:attribute>
<xs:simpleType name="aggregateType">
  <xs:restriction base="xs:normalizedString">
    <xs:enumeration value="array"/>
    <xs:enumeration value="list"/>
    <xs:enumeration value="set"/>
    <xs:enumeration value="bag"/>
    <xs:enumeration value="array-unique"/>
    <xs:enumeration value="array-optional"/>
    <xs:enumeration value="array-optional-unique"/>
    <xs:enumeration value="list-unique"/>
  </xs:restriction>
</xs:simpleType>
<xs:attributeGroup name="instanceAttributes">
  <xs:attribute name="id" type="xs:ID" use="optional"/>
  <xs:attribute name="path" type="xs:NMTOKENS" use="optional"/>
  <xs:attribute name="pos" use="optional">
    <xs:simpleType>
      <xs:restriction>
        <xs:simpleType>
          <xs:list itemType="xs:integer"/>
        </xs:simpleType>
        <xs:minLength value="1"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:attributeGroup>
<xs:attribute name="attributeType">
  <xs:simpleType>
    <xs:restriction base="xs:normalizedString">

```

```

        <xs:enumeration value="explicit"/>
        <xs:enumeration value="inverse"/>
        <xs:enumeration value="derived"/>
        <xs:enumeration value="renamed"/>
    </xs:restriction>
</xs:simpleType>
</xs:attribute>
<!-- SIMPLE TYPE DEFINITIONS -->
<xs:complexType name="base64Binary">
    <xs:simpleContent>
        <xs:extension base="xs:base64Binary">
            <xs:attribute name="extraBits" type="xs:integer"
use="optional"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="hexBinary">
    <xs:simpleContent>
        <xs:extension base="xs:hexBinary">
            <xs:attribute name="extraBits" type="xs:integer"
use="optional"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:simpleType name="logical">
    <xs:restriction base="xs:string">
        <xs:enumeration value="false"/>
        <xs:enumeration value="true"/>
        <xs:enumeration value="unknown"/>
    </xs:restriction>
</xs:simpleType>
<!-- OTHER NEEDED TYPE DEFINITIONS -->
<xs:simpleType name="Seq-anyURI">
    <xs:list itemType="xs:anyURI"/>
</xs:simpleType>
<!-- WRAPPER FOR ATOMIC SIMPLE TYPES -->
<xs:element name="base64Binary-wrapper" nillable="true">
    <xs:complexType>
        <xs:simpleContent>
            <xs:extension base="exp:base64Binary">
                <xs:attributeGroup ref="exp:instanceAttributes"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:element>
<xs:element name="hexBinary-wrapper" nillable="true">
    <xs:complexType>
        <xs:simpleContent>
            <xs:extension base="exp:hexBinary">
                <xs:attributeGroup ref="exp:instanceAttributes"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:element>
<xs:element name="integer-wrapper" nillable="true">
    <xs:complexType>
        <xs:simpleContent>
            <xs:extension base="xs:integer">
                <xs:attributeGroup ref="exp:instanceAttributes"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:element>

```

```

        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:element name="logical-wrapper" nillable="true">
    <xs:complexType>
        <xs:simpleContent>
            <xs:extension base="exp:logical">
                <xs:attributeGroup ref="exp:instanceAttributes"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:element>
<!-- WRAPPER FOR ATOMIC SIMPLE TYPES -->
<xs:element name="boolean-wrapper" nillable="true">
    <xs:complexType>
        <xs:simpleContent>
            <xs:extension base="xs:boolean">
                <xs:attributeGroup ref="exp:instanceAttributes"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:element>
<xs:element name="double-wrapper" nillable="true">
    <xs:complexType>
        <xs:simpleContent>
            <xs:extension base="xs:double">
                <xs:attributeGroup ref="exp:instanceAttributes"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:element>
<xs:element name="decimal-wrapper" nillable="true">
    <xs:complexType>
        <xs:simpleContent>
            <xs:extension base="xs:decimal">
                <xs:attributeGroup ref="exp:instanceAttributes"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:element>
<xs:element name="long-wrapper" nillable="true">
    <xs:complexType>
        <xs:simpleContent>
            <xs:extension base="xs:long">
                <xs:attributeGroup ref="exp:instanceAttributes"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:element>
<xs:element name="string-wrapper" nillable="true">
    <xs:complexType>
        <xs:simpleContent>
            <xs:extension base="xs:normalizedString">
                <xs:attributeGroup ref="exp:instanceAttributes"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>

```

```

</xs:element>
<xs:element name="generalString-wrapper" nillable="true">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attributeGroup ref="exp:instanceAttributes"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="language-wrapper" nillable="true">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:language">
        <xs:attributeGroup ref="exp:instanceAttributes"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="Name-wrapper" nillable="true">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:Name">
        <xs:attributeGroup ref="exp:instanceAttributes"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="QName-wrapper" nillable="true">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:QName">
        <xs:attributeGroup ref="exp:instanceAttributes"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="NMTOKEN-wrapper" nillable="true">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:NMTOKEN">
        <xs:attributeGroup ref="exp:instanceAttributes"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="anyURI-wrapper" nillable="true">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:anyURI">
        <xs:attributeGroup ref="exp:instanceAttributes"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
</xs:schema>

```

## Schematron Obtido

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Schematron file created by EXP2SCH Tool version 0.2, written by Sergio Onofre
(UNINOVA) smo@uninova.pt-->
<!--Schema derived from null-->
<schema xmlns="http://www.ascc.net/xml/schematron">
  <pattern name="Address - WR1">
    <rule context="Address">
      <assert test="(StateOrProvinceCode) or (not((CountryCode = 'US') or
(CountryCode = 'CA')))) or @ref">ERROR
    </assert>
  </rule>
</pattern>
<pattern name="Address - WR2">
  <rule context="Address">
    <assert test="not((Units/LenghtUnit)) or ((Length_) or ((Width)) or
((Height))) or @ref">ERROR
  </assert>
</rule>
</pattern>
<pattern name="Calendar_date - WR1">
  <rule context="Calendar_date">
    <assert test="javaf:ValidCalendarDate(current()) or @ref">ERROR
  </assert>
</rule>
</pattern>
<pattern name="DeclaredValue - WR1">
  <rule context="DeclaredValue">
    <assert test="not((Value_) or ((CurrencyCode) and (CurrencyCode =
'USD'))) or @ref">ERROR
  </assert>
</rule>
</pattern>
<pattern name="OriginAddress- WR1">
  <rule context="OriginAddress">
    <assert test="(StateOrProvinceCode) or (not((CountryCode = 'US') or
(CountryCode = 'CA')))) or @ref">ERROR
  </assert>
</rule>
</pattern>
<pattern name="OriginAddress- WR2">
  <rule context="OriginAddress">
    <assert test="(PostalCode) or (not((CountryCode = 'US') or
(CountryCode = 'CA')))) or @ref">ERROR
  </assert>
</rule>
</pattern>
<pattern name="Dimensions - WR1">
  <rule context="Dimensions">
    <assert test="not((Units)) or ((Length_) or ((Width)) or ((Height))) or
@ref">ERROR
  </assert>
</rule>
</pattern>
<pattern name="FDXRateRequest - WR1">
  <rule context="FDXRateRequest">
```

```

        <assert test="((Service != 'INTERNATIONALPRIORITY') and
        (Service != 'INTERNATIONALECONOMY') and (Service !=
        'INTERNATIONALFIRST') and (Service != 'INTERNATIONALPRIORITY_FREIGHT') and
        (Service != 'INTERNATIONALECONOMY_FREIGHT') and (Service !=
        'EUROPEFIRSTINTERNATIONALPRIORITY')) or ((DropoffType = 'REGULARPICKUP') or
        (DropoffType = 'REQUESTCOURIER') or (DropoffType = 'STATION')) or @ref">ERROR
        </assert>
    </rule>
</pattern>
<pattern name="FDXRateRequest - WR2">
    <rule context="FDXRateRequest">
        <assert test="not(PackageCount) or (0 &lt; PackageCount and
        PackageCount &lt; 1000) or @ref">ERROR
        </assert>
    </rule>
</pattern>
<pattern name="DestinationAddress- WR1">
    <rule context="Address">
        <assert test="(StateOrProvinceCode) or (not((CountryCode = 'US') or
        (CountryCode = 'CA')))) or @ref">ERROR
        </assert>
    </rule>
</pattern>
<pattern name="DestinationAddress- WR2">
    <rule context="Address">
        <assert test="(PostalCode) or (not((CountryCode = 'US') or
        (CountryCode = 'CA')))) or @ref">ERROR
        </assert>
    </rule>
</pattern>
<pattern name="RequestHeader - UR1">
    <rule context="RequestHeader">
        <report test=
        "not(count(//RequestHeader/CustomerTransactionIdentifier[.
        =current()/CustomerTransactionIdentifier])&lt;=1) and not(@ref)">ERROR
        </report>
    </rule>
</pattern>
<pattern name="RequestHeader - UR2">
    <rule context="RequestHeader">
        <report test="not(count(//RequestHeader/MeterNumber[. =
        current()/MeterNumber])&lt;=1) and not(@ref)">ERROR
        </report>
    </rule>
</pattern>
<pattern name="VariableHandlingCharges - WR1">
    <rule context="VariableHandlingCharges">
        <assert test="not((Type_)) or ((Level)) or @ref">ERROR
        </assert>
    </rule>
</pattern>
<pattern name="VariableHandlingCharges - WR2">
    <rule context="VariableHandlingCharges">
        <assert test="not((Type_)) or ((AmountOrPercentage)) or
        @ref">ERROR
        </assert>
    </rule>
</pattern>
</schema>

```

## Java Obtido

```
import org.w3c.dom.*;

public class JavaFunction
{
    public Node findNode(String name, Node n) throws Exception
    {
        try
        {
            NodeList nl = n.getChildNodes();
            int i = 0;
            Node out = null;
            while (i != nl.getLength())
            {
                Node a = nl.item(i);
                if (a.getLocalName() == name)
                {
                    out = a;
                }
                i++;
            }
            return out;
        } catch (Exception e)
        {
            return null;
        }
    }

    public boolean LeapYear(Node Year) throws Exception
    {
        try
        {
            if (Integer.parseInt(Year.getFirstChild().getNodeValue()) % (4) == 0      &&
                (Integer.parseInt(Year.getFirstChild().getNodeValue()) % (100) != 0) ||
                (Integer.parseInt(Year.getFirstChild().getNodeValue()) % (400) == 0))
            {
                return true;
            }
            else
            {
                return false;
            }
        } catch (Exception e)
        {
            return false;
        }
    }

    public boolean ValidCalendarDate(Node Date) throws Exception
    {
        try
        {
            if (!(1 <= Integer.parseInt(findNode("day_component",
                Date).getFirstChild().getNodeValue()) &&
                Integer.parseInt(findNode("day_component",
                Date).getFirstChild().getNodeValue()) <= 31))
            {
                return false;
            }
        }
    }
}
```

```

    }
    switch (Integer.parseInt(findNode("month_component",
Date).getFirstChild().getNodeValue()))
    {
        case 4: return 1 <=
Integer.parseInt(findNode("day_component",
Date).getFirstChild().getNodeValue()) &&
Integer.parseInt(findNode("day_component",
Date).getFirstChild().getNodeValue()) <= 30;
        case 6: return 1 <=
Integer.parseInt(findNode("day_component",
Date).getFirstChild().getNodeValue()) &&
Integer.parseInt(findNode("day_component",
Date).getFirstChild().getNodeValue()) <= 30;
        case 9: return 1 <=
Integer.parseInt(findNode("day_component",
Date).getFirstChild().getNodeValue()) &&
Integer.parseInt(findNode("day_component",
Date).getFirstChild().getNodeValue()) <= 30;
        case 11: return 1 <=
Integer.parseInt(findNode("day_component",
Date).getFirstChild().getNodeValue()) &&
Integer.parseInt(findNode("day_component",
Date).getFirstChild().getNodeValue()) <= 30;
        case 2: if (LeapYear(findNode("year_component", Date)))
        {
            return 1 <=
Integer.parseInt(findNode("day_component",
Date).getFirstChild().getNodeValue()) &&
Integer.parseInt(findNode("day_component",
Date).getFirstChild().getNodeValue()) <= 29;
        }
        else
        {
            return 1 <=
Integer.parseInt(findNode("day_component",
Date).getFirstChild().getNodeValue()) &&
Integer.parseInt(findNode("day_component",
Date).getFirstChild().getNodeValue()) <= 28;
        }
        default: return true;
    }
}
catch (Exception e)
{
    return false;
}
}}

```