



Universidade Nova de Lisboa  
Faculdade de Ciências e Tecnologia

**DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA**

**INTEGRAÇÃO DE AMBIENTES INDUSTRIAIS COLABORATIVOS ATRAVÉS DA  
VISUALIZAÇÃO ORIENTADA PELOS MODELOS**

POR:

**BRUNO MIGUEL PEREIRA DE ALMEIDA**

DISSERTAÇÃO APRESENTADA NA FACULDADE DE CIÊNCIAS E TECNOLOGIA DA UNIVERSIDADE NOVA DE LISBOA  
PARA OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES

**ORIENTADOR: DOUTOR RICARDO LUÍS ROSA JARDIM GONÇALVES**

**LISBOA**

**2009**



## 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.

Em primeiro lugar, gostaria de deixar um agradecimento especial aos meus pais e aos meus avós maternos, em especial ao meu avô que faleceu no passado ano de 2007 e gostaria de ter visto este trabalho concluído, por todo o apoio que sempre me deram e por terem feito sempre tudo para eu poder chegar até aqui.

Gostaria também de agradecer aos meus amigos, em especial à Inês Lino, por todo o apoio que me tem dado e ao João Águas pelas noitadas que fizemos juntos durante todo o curso e que fizeram este trabalho acontecer.

Quero também agradecer muito em especial ao Mestre Pedro Maló por sempre ter acreditado em mim e desde cedo ter-me incentivado para a investigação, estimulando-me com um misto de mestria e amizade a encarar o início da minha vida profissional e a fechar este capítulo importante de longos anos de estudo.

Agradeço também ao grupo de investigação GRIS por sempre me terem recebido bem, e por me terem apoiado no desenrolar deste trabalho, em especial ao Mestre Fernando Ferreira.

Por último, agradeço ao meu orientador Prof. Ricardo Jardim Gonçalves por me ter possibilitado fazer este trabalho no âmbito de um projecto internacional de investigação e, por sempre me ter orientado nos momentos certos e por me ter permitido uma liberdade de acção que foi decisiva para que este trabalho contribuísse para o meu desenvolvimento pessoal.

---



## SUMÁRIO

Para um produto ser bem sucedido no contexto económico actual, caracterizado por ser dinâmico, global e em constante mudança, é importante que este tenha um preço competitivo, traga um significativo valor acrescentado e consiga um tempo reduzido de entrada no mercado. Para tal, novas formas organizacionais têm sido utilizadas, suportadas por equipas de desenvolvimento colaborativas, especializadas e distribuídas. Contudo, neste contexto, surgem frequentemente dificuldades na interoperabilidade entre aplicações e sistemas, onde, com um inúmero volume de informação a circular, aparecem obstáculos na interpretação e utilização da informação no trabalho específico de cada interveniente. Para se reduzir estas dificuldades, têm-se investigado técnicas como a da visualização. Nesta dissertação, estas são propostas num âmbito de aplicação em empresas suportadas por modelos, levando à integração dos vários intervenientes no ciclo de vida do produto. A solução proposta combina a aplicação de vistas sobre dados, a harmonização de formatos de troca de dados e a contextualização da informação para promover uma melhor interoperabilidade entre aplicações e sistemas em ambientes colaborativos. Esta baseia-se no paradigma do desenvolvimento orientado pelos modelos, em particular nos morfismos entre modelos, como base para a junção destas características numa abordagem interoperável.

---

## ABSTRACT

For a product to be successful in the current economic context, characterized by be dynamic, global and rapidly changing, it is important that it have a competitive price, bring high value, and has a low time to market. For that, new organizational forms are being used, supported by collaborative, specialized and distributed development teams. Although in this context, early difficulties arise in interoperability between applications and systems. Here, with the endless amount of information exchanged, new obstacles appear in the interpretation and utilization of that information in the specific work of each actor. To overcome these difficulties, research has been made on techniques like visualisation. In this dissertation, these techniques are proposed on a scope of model based enterprises in order to integrate the various actors of the product life cycle. The proposed solution combines the application of views over data, the harmonization of data exchange formats, and contextualization of information for better interoperability between application and systems in collaborative environments. This is based on the paradigm of model driven development, particularly in model morphisms as a basis for the union of these characteristics in an interoperable approach.

---

---

## SIMBOLOGIA E NOTAÇÕES

- ANSI** *PT*: Instituto nacional americano de normalização  
*EN*: American National Standards Institute  
*Ref.*: <http://www.ansi.org/>
- ATL** *PT*: Linguagem de transformação ATLAS  
*EN*: ATLAS Transforming Language  
*Ref.*: <http://www.eclipse.org/m2m/atl/>
- CAD** *PT*: Concepção Assistida por Computador  
*EN*: Computer-aided Design
- CAE** *PT*: Engenharia Assistido por Computador  
*EN*: Computer-aided Engineering
- CAM** *PT*: Manufatura Assistido por Computador  
*EN*: Computer-aided Manufacturing
- CEN** *PT*: Comité Europeu para a Normalização  
*FR*: Comité Européen de Normalisation  
*Ref.*: <http://www.cen.eu/>
- ECMA** *PT*: Associação Europeia para a Manufatura Assistida por Computador  
*EN*: European Computer Manufacturers Association  
*Ref.*: <http://www.ecma-international.org/>
- ICT** *PT*: Tecnologias de Informação e Comunicação  
*EN*: Information and Communication Technologies  
*Ref.*: <http://cordis.europa.eu/fp7/ict/>
- IDE** *PT*: Ambiente Integrado de Desenvolvimento  
*EN*: Integrated Development Environment
- IEC** *PT*: Comissão Internacional de Electrotecnia  
*EN*: International Electrotechnical Commission  
*Ref.*: <http://www.iec.ch/>
-

- IEEE** *PT*: Instituto de Engenheiros Electrotécnicos e Electrónicos  
*EN*: Institute of Electrical and Electronics Engineers  
*Ref.*: <http://www.ieee.org/>
- ISO** *PT*: Organização Internacional para a Normalização  
*EN*: International Organization for Standardization  
*Ref.*: <http://www.iso.org/>
- JVM** *PT*: Máquina virtual do Java  
*EN*: Java Virtual Machine
- MBE** *PT*: Organização Suportada em Modelos  
*EN*: Model Based Enterprise
- MDD** *PT*: Desenvolvimento Orientado pelos Modelos  
*EN*: Model Driven Development
- MF** *PT*: Campo de valor múltiplo  
*EN*: Multiple-valued Field
- MoMo** *PT*: Morfismo de Modelo  
*EN*: Model Morphism
- NIST** *PT*: Instituto Nacional para Normas e Tecnologia (EUA)  
*EN*: National Institute of Standards and Technology (USA)  
*Ref.*: <http://www.nist.gov/>
- ODA** *PT*: Aliança para a Concepção Aberta  
*EN*: Open Design Alliance  
*Ref.*: <http://www.opendwg.org/>
- OMG** *PT*: Grupo para a gestão de objectos  
*EN*: Object Management Group  
*Ref.*: <http://www.omg.org/>
- PDMA** *PT*: Associação para o desenvolvimento e gestão de produtos  
*EN*: Product Development and Management Association  
*Ref.*: <http://www.pdma.org/>
-



- 
- R&D** *PT*: Investigação & Desenvolvimento  
*EN*: Research & Development
- SaaS** *PT*: Software como um serviço  
*EN*: Software as a Service
- SF** *PT*: Campo de valor único  
*EN*: Single-valued Field
- STEP** *PT*: Standard para a troca de informação de produtos (ISO 10303)  
*EN*: Standard for the Exchange of Product model data (ISO 10303)  
*Ref.*: <http://www.tc184-sc4.org/>
- TTCN** *PT*: Notação tabular e de árvore combinada  
*EN*: The tree and Tabular Combined Notation  
*Ref.*: <http://www.ttcn-3.org/>
- UML** *PT*: Linguagem de Modelação Unificada  
*EN*: Unified Modeling Language  
*Ref.*: <http://www.uml.org/>
- VRML** *PT*: Linguagem para modelação de realidade virtual  
*EN*: Virtual Reality Modeling Language  
*Ref.*: <http://www.web3d.org/x3d/vrml/>
- W3C** *PT*: Consórcio para a World Wide Web  
*EN*: The World Wide Web Consortium  
*Ref.*: <http://www.w3.org/>
- X3D** *PT*: 3D Extensível  
*EN*: eXtensible 3D  
*Ref.*: <http://www.web3d.org/x3d/>
- XML** *PT*: Linguagem de marcação extensível  
*EN*: eXtensible Markup Language  
*Ref.*: <http://www.w3.org/XML/>
-



# ÍNDICE DE MATÉRIAS

<b>1. INTRODUÇÃO .....</b>	<b>17</b>
1.1. MOTIVAÇÃO.....	21
1.2. CONTEXTO .....	23
1.3. METODOLOGIA DE INVESTIGAÇÃO.....	25
1.4. ESTRUTURA DE CAPÍTULOS .....	27
<b>2. VISUALIZAÇÃO DE DADOS DE PRODUTO EM AMBIENTES INDUSTRIAIS .....</b>	<b>29</b>
2.1. VISUALIZAÇÃO NO CICLO DE VIDA DO PRODUTO .....	30
2.1.1. <i>Impactos da visualização</i> .....	31
2.1.2. <i>Desafios para a aplicação da visualização</i> .....	34
2.2. NORMAS DE TROCA DE DADOS PARA A VISUALIZAÇÃO .....	36
2.2.1. <i>Características relevantes</i> .....	36
2.2.2. <i>Normas de referência</i> .....	39
2.2.3. <i>Análise comparativa</i> .....	42
<b>3. VISUALIZAÇÃO SUPOSTADA EM MODELOS .....</b>	<b>45</b>
3.1. DESENVOLVIMENTO ORIENTADO PELOS MODELOS.....	46
3.1.1. <i>Modelos, meta-modelos e linguagens de modelação</i> .....	47
3.1.2. <i>Morfismos de modelos</i> .....	50
3.2. VISUALIZAÇÃO ORIENTADA PELOS MODELOS .....	55
3.2.1. <i>Vistas orientadas pelos modelos</i> .....	56
<b>4. SOLUÇÃO INTEROPERÁVEL PARA A VISUALIZAÇÃO DE DADOS DE PRODUTO .....</b>	<b>59</b>
4.1. FUNDAMENTOS DE SUPORTE À SOLUÇÃO .....	60
4.2. CONCEITO DA SOLUÇÃO .....	61
4.3. ARQUITECTURA DA SOLUÇÃO .....	64
4.3.1. <i>Aplicação de modelos</i> .....	64
4.3.2. <i>Morfismos de modelos</i> .....	66
4.3.3. <i>Arquitectura geral</i> .....	69
<b>5. IMPLEMENTAÇÃO DA SOLUÇÃO .....</b>	<b>71</b>
5.1. ESCOLHAS TECNOLÓGICAS .....	71
5.1.1. <i>Linguagem de programação</i> .....	72
5.1.2. <i>Estrutura de dados da base comum</i> .....	73
5.1.3. <i>Formatos geométricos</i> .....	74
5.1.4. <i>Vistas</i> .....	75
5.2. INSTANCIACÃO TECNOLÓGICA À ARQUITECTURA.....	76
5.3. PROCESSO DE IMPLEMENTAÇÃO .....	77

---

5.3.1.	<i>Base comum</i> .....	77
5.3.2.	<i>Ligação base comum com formatos geométricos</i> .....	78
5.3.3.	<i>Ligação base comum às vistas</i> .....	79
5.3.4.	<i>Aplicação da contextualização da informação</i> .....	80
5.4.	INTERFACE DE UTILIZAÇÃO .....	80
<b>6.</b>	<b>TESTES E VALIDAÇÃO</b> .....	<b>81</b>
6.1.	METODOLOGIA PARA TESTES.....	81
6.1.1.	<i>ISO/IEC 9646 – Plataforma e metodologia para testes de conformidade</i> .....	81
6.1.2.	<i>Notação para testes – TTCN – Notação Tabular e de Árvore Combinada</i> .....	83
6.2.	TESTES FUNCIONAIS .....	84
6.2.1.	<i>Testes da harmonização dos formatos</i> .....	85
6.2.2.	<i>Testes da personalização de vistas</i> .....	87
6.2.3.	<i>Testes da contextualização da informação</i> .....	89
6.3.	VEREDICTO ACERCA DA CONFORMIDADE DA SOLUÇÃO .....	91
<b>7.</b>	<b>CONCLUSÕES E PERSPECTIVAS FUTURAS</b> .....	<b>93</b>
<b>8.</b>	<b>BIBLIOGRAFIA</b> .....	<b>97</b>
<b>ANEXO A.</b>	<b>NORMAS PARA A REPRESENTAÇÃO GEOMÉTRICA</b> .....	<b>103</b>
<b>ANEXO B.</b>	<b>MORFISMOS ENTRE O DWG E A BASE COMUM EM X3D</b> .....	<b>115</b>
B.1.	TIPOS BÁSICOS .....	115
B.2.	TIPOS COMPOSTOS (OBJECTOS GEOMÉTRICOS).....	118
<b>ANEXO C.</b>	<b>EXECUÇÃO DOS TESTES</b> .....	<b>125</b>
C.1.	TESTES DA HARMONIZAÇÃO DOS FORMATOS .....	125
C.1.1.	<i>Teste de morfismo – Elementos gráficos simples</i> .....	125
C.1.2.	<i>Teste de morfismo – Desenho composto</i> .....	133
C.2.	TESTES DA PERSONALIZAÇÃO DE VISTAS .....	135
C.2.1.	<i>Visualizações – Visualizador tridimensional</i> .....	135
C.2.2.	<i>Visualizações – Vista tabular</i> .....	135
C.3.	TESTES DA CONTEXTUALIZAÇÃO DA INFORMAÇÃO.....	137

---

---

## ÍNDICE DE FIGURAS

Figura 1.1 – Distribuição dos intervenientes no ciclo de vida do produto. ....	18
Figura 1.2 – Metodologia de investigação. ....	25
Figura 2.1 – Ciclo de vida do produto vs actividades estudadas no impacto da visualização. ....	31
Figura 2.2 – Relação entre a largura de banda utilizada e compressão da informação face à precisão da informação. ....	37
Figura 2.3 – Gráfico comparativo das normas de representação estudadas.....	44
Figura 3.1 – Relação entre modelos, meta-modelos e linguagens de modelação. ....	47
Figura 3.2 – Arquitectura de modelação segundo a OMG.....	48
Figura 3.3 – Operações de refinamento e abstracção em modelos. ....	49
Figura 3.4 – Morfismo de modelos. ....	50
Figura 3.5 – Diferentes tipos de correspondências entre dois modelos.....	52
Figura 3.6 – Transformação orientada pelos meta-modelos.....	54
Figura 3.7 – Arquitectura de referência para a visualização orientada pelos modelos adaptada de (Bull, Storey, Favre, & Litoiu, 2006).....	56
Figura 3.8 – Aplicação da visualização orientada por modelos a uma árvore e uma lista. ....	58
Figura 4.1 – Sistema para unificação da visualização de dados de produto.....	61
Figura 4.2 – Demonstração do número de ligações entre formatos e artefactos visuais. ....	62
Figura 4.3 – Especificação da solução. ....	63
Figura 4.4 – Formato específico orientado pelos modelos.....	65
Figura 4.5 – Vistas orientadas pelos modelos.....	66
Figura 4.6 – Morfismo entre o formato específico e a base comum.....	67
Figura 4.7 – Morfismo entre a base comum e a visualização. ....	68
Figura 4.8 – Morfismos na aplicação das regras de contextualização da informação.....	69
Figura 4.9 – Arquitectura para a unificação da visualização de dados de produto. ....	70
Figura 5.1 – Funcionamento da máquina virtual do java (Sun Microsystems, Inc.) .....	72

---

---

Figura 5.2 – X3D aplicado ao desenvolvimento orientado pelos modelos. ....	74
Figura 5.3 – Tabela descrita em UML.....	76
Figura 5.4 – Aplicação das escolhas tecnológicas à arquitectura proposta. ....	76
Figura 5.5 – Arquitectura do JAXB. ....	77
Figura 5.6 – Exemplo de transformação obtida com a utilização do JAXB.....	78
Figura 5.7 – Exemplo de transformação de objecto DWG em classes java.....	79
Figura 5.8 – Interface gráfica da aplicação desenvolvida para a prova de conceito.....	80
Figura 6.1 – Vista global do processo de teste de conformidade.....	82
Figura A.1 – Desenho a representar. ....	107
Figura A.2 – Ficheiro IGES Resultante.....	107
Figura A.3 – Resultado de uma ressonância magnética a uma laranja. ....	110
Figura A.4 – Exemplo da decomposição de um objecto em triângulos.....	111
Figura A.5 – Fragmento de um ficheiro STL, no modo ASCII. ....	112
Figura A.6 – Exemplo de um ficheiro de formato X3D. ....	113
Figura A.7 – Representação 3D do objecto demonstrado na Figura A.6.....	114
Figura B.1 – Ordem dos vértices utilizados na descrição de uma face tridimensional. ....	121
Figura B.2 – Vertices utilizados para definir as faces no objecto sólido.....	123
Figura C.1 – Desenho com elementos geometricos simples a ser testado. ....	126
Figura C.2 – Desenho obtido através da utilização do X3D exportado. ....	127
Figura C.3 – Definição do objecto tipo arco no AutoCAD.....	128
Figura C.4 – Resultado da exportação do objecto tipo arco para X3D.....	128
Figura C.5 – Definição do objecto tipo círculo no AutoCAD. ....	129
Figura C.6 – Resultado da exportação do objecto tipo arco para X3D.....	129
Figura C.7 – Definição do objecto tipo linha no AutoCAD.....	130
Figura C.8 – Resultado da exportação do objecto tipo linha para X3D.....	130
Figura C.9 – Definição do objecto tipo face tridimensional no AutoCAD.....	131

---

Figura C.10 – Resultado da exportação do objecto face tridimensional para X3D.....	131
Figura C.11 – Definição do objecto tipo sólido no AutoCAD.....	132
Figura C.12 – Resultado da exportação do objecto tipo sólido para X3D.....	132
Figura C.13 – Desenho composto a ser testado.....	134
Figura C.14 – Desenho resultante da exportação. ....	134
Figura C.15 – Vista tabular obtida. ....	136
Figura C.16 – Exportação do ficheiro sem aplicação de regras de contextualização.....	138
Figura C.17 – Exportação do ficheiro com aplicação de regras de contextualização. ....	140

---

---

## ÍNDICE DE TABELAS

Tabela 2.1 – Características das várias normas de representação geométrica.....	43
Tabela 6.1 – Exemplo de teste definido em TTCN. ....	84
Tabela 6.2 – Detalhe do teste de morfismo – Elementos gráficos simples.....	86
Tabela 6.3 – Detalhe do teste de morfismo – Desenho composto. ....	87
Tabela 6.4 – Detalhe do teste de visualizações – Visualizador tridimensional. ....	88
Tabela 6.5 – Detalhe do teste de visualizações – Vista tabular.....	89
Tabela 6.6 – Detalhe do teste da contextualização da informação.....	90
Tabela B.1 – Mapeamento do objecto tipo arco. ....	119
Tabela B.2 – Mapeamento do objecto tipo circulo. ....	120
Tabela B.3 – Mapeamento do objecto tipo linha. ....	121
Tabela B.4 – Mapeamento do objecto tipo face tridimensional. ....	122
Tabela B.5 – Mapeamento do objecto tipo sólido.....	123

---



## 1. INTRODUÇÃO

A pressão, dinamismo e competitividade dos mercados ditam uma necessidade cada vez maior de redução dos tempos de entrada dos produtos no mercado, de menores preços finais e de produtos de maior valor acrescentado. Em virtude deste facto, surgiram novas formas organizacionais (e.g. empresas virtuais, empresas estendidas, etc.) onde as interacções se regem por padrões crescentes de colaboração e suportadas em tecnologias de informação e comunicação (ICT, do inglês, *Information and Communication Technologies*), o que tem permitindo responder de uma forma mais efectiva às necessidades e desafios, através da integração de todas as fases do ciclo de vida do produto.

Uma colaboração eficiente entre os intervenientes durante todo o ciclo de vida do produto, abrangendo concepção, engenharia, aquisições, produção, administração e serviços torna-se um factor chave para o sucesso. Todas os agentes envolvidos no ciclo de vida do produto (desde consumidores, projectistas, engenheiros, técnicos de produção, operadores a vendedores), necessitam da informação apropriada no momento certo e da forma correcta, para se poder actuar rapidamente e acrescentar valor ao desempenho da sua função (Schilli & Dai, 2006).

No entanto, e apesar do mercado se apresentar cada vez mais exigente e a necessidade de combinar esforços mais relevante, a pragmática da colaboração no desenvolvimento de produtos não é um tema novo. Por exemplo, em 1998, S.S.A. Willaert, R. de Graaf e S. Minderhoud visionaram a Engenharia Colaborativa, definindo-a como “uma abordagem sistemática ao controlo de custos do ciclo de vida, à qualidade do produto e ao tempo de entrada no mercado, desenvolvendo-se concorrentemente os produtos e os seus processos respondendo às expectativas do consumidor, onde a capacidade de decisão é assegurada pela entrada de informação e avaliação por parte de todas as disciplinas envolvidas no ciclo de vida do produto, incluindo fornecedores, e aplicando as tecnologias de informação para suportar a troca de informação quando necessária” (Willaert, Graaf, & Minderhoud, 1998).

A engenharia colaborativa estabelece que redes, englobando diferentes empresas ou unidades distintas dentro duma mesma organização, reúnam as suas competências multidisciplinares, de forma a facilitar e melhorar o trabalho de concepção e engenharia, processos de produção, e serviços de operação e manutenção, e também actividades de

---

promoção e distribuição. A colaboração tanto pode acontecer dentro de um mesmo espaço físico, ser distribuída por várias equipas dispersas geograficamente ou ainda ser estendida por organizações externas de forma a aumentar a competitividade e capacidade de resposta. Esta abordagem aumenta a flexibilidade da organização para responder aos mais complexos e distintos requisitos dos utilizadores (Hammond, Koubek, & Harvey, 2001).

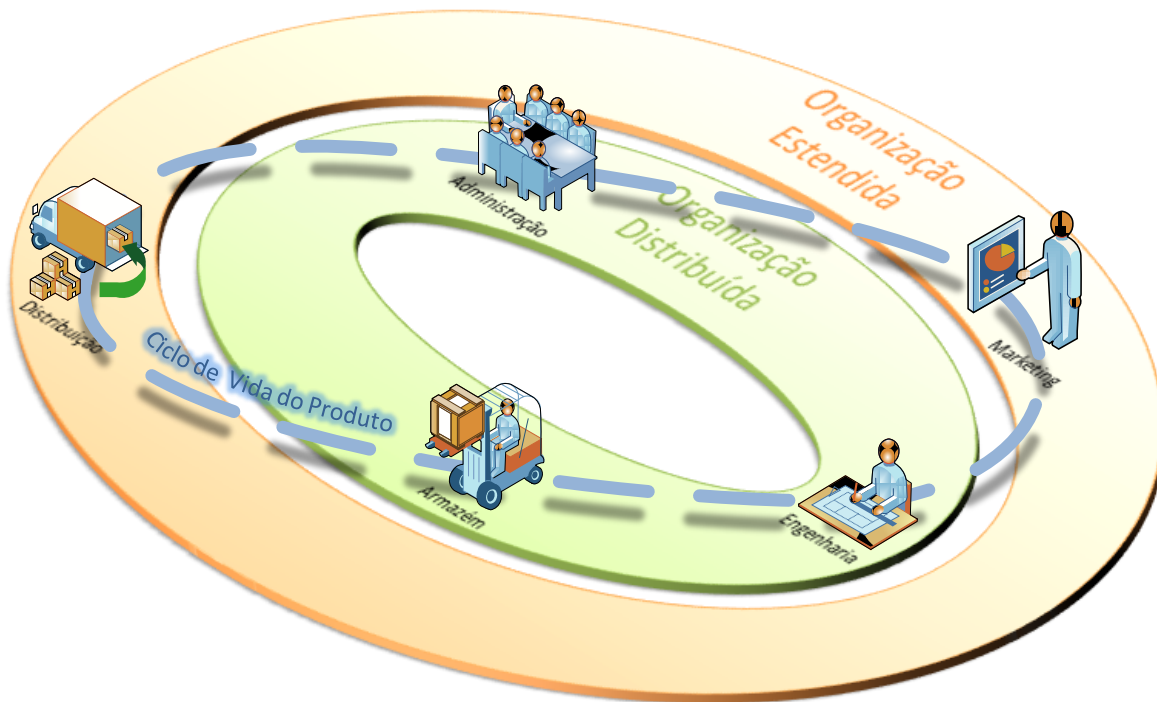


FIGURA 1.1 – DISTRIBUIÇÃO DOS INTERVENIENTES NO CICLO DE VIDA DO PRODUTO.

As tecnologias de informação e comunicação criam novas oportunidades para uma colaboração mais efectiva entre todos os intervenientes no ciclo de vida do produto. Exemplo disto foi a introdução da concepção assistida por computador, em inglês *Computer Aided Design* (CAD) que também deu a sua contribuição para essa eficácia. Através deste, e com a evolução das novas tecnologias outras disciplinas surgiram e contribuíram ainda mais para um aumento da eficácia, sendo as mais sonantes a Engenharia Assistida por Computador (CAE) e Manufatura Assistida por Computador (CAM).

O CAD tem-se tornado imprescindível no ciclo de vida do produto, especialmente na fase de desenvolvimento, onde estes são concebidos, as suas componentes e os desenhos técnicos dos mesmos. A aplicação do CAE permite depois o acesso a métodos de ensaio virtuais, tais como simulações e testes recorrendo-se por exemplo, a métodos numéricos tais como CFD – Dinâmica de fluidos computacional, permitindo, assim, antecipar problemas e testar

soluções alternativas de concepção. Finalmente, o CAM foca-se na manufactura de peças reais, tanto protótipos dos produtos, recorrendo a uma das técnicas em voga na indústria, a prototipagem rápida, como à produção de ferramentas específicas para a produção dos produtos em escala industrial (Luo, Lan, Tzou, & Lin, 2004).

No entanto, estas disciplinas só por si não trazem uma mais-valia ao ciclo de vida do produto, necessitando, portanto, de estarem incluídas no âmbito do trabalho colaborativo para terem uma maior relevância. Apesar de estes sistemas serem utilizados num paradigma de trabalho colaborativo, tais nem sempre estão preparados para o efeito. Uma das barreiras existentes para o uso destes num ambiente verdadeiramente colaborativo advem da situação de cada vendedor promover um formato proprietário de troca de informação, resultando na dificuldade de troca de informação entre sistemas heterogéneos.

Com estas dificuldades, cria-se a necessidade de normalizar as trocas de informação. A normalização permite que se definam bases comuns de troca de informação, passível de ser utilizadas entres parceiros e grupos industriais sem a existência das dificuldades referidas. Mesmo existindo formas normalizadas para a troca de informação é necessário a aceitação dessas normas pelos vendedores de software, de forma a introduzirem o suporte às mesmas nos seus produtos.

Uma primeira abordagem a este problema foi, em 1979, a união de um grupo de vendedores e utilizadores de software CAD, constituído por empresas de renome como a Boeing e a General Electric, e suportado pelo NIST e pelo Departamento de Defesa dos Estados Unidos, que iniciaram um projecto, IGES *Initial Graphics Exchange Specification* (NIST, 2002), em português, “Especificação inicial para a troca gráfica”, que visava a normalização da troca de modelos gráficos criados em software de diferentes vendedores. Hoje em dia a grande maioria dos sistemas CAD permitem a exportação e importação de modelos neste formato.

Do ponto de vista económico, a eficiência e eficácia na troca de informação é relevante, porque todas as barreiras existentes na troca de informação acarretam custos. Segundo um estudo do NIST datado de 2004 (Gallaher, O'Connor, Dettbarn Jr., & Gilday, 2004), na construção de obras públicas dos Estados Unidos da América a deficiência na troca de informação entre aplicações é estimada num custo anual na casa dos 15,8 mil milhões de dólares.

---

Mas as dificuldades de se efectuar trabalho colaborativo não acabam com a normalização dos formatos de troca de dados, é que mesmo as aplicações promovidas pelos mesmos vendedores podem contar com dificuldades em ambientes de trabalho colaborativo. Por exemplo, na indústria da aeronáutica o desenvolvimento do famoso avião Airbus© A380<sup>1</sup> contou com o desenvolvimento distribuído em vários países e várias empresas recorrendo-se a avançados sistemas informáticos, nomeadamente o CATIA<sup>2</sup> da Dassault Systèmes. Devido a uma actualização na versão do software utilizado em França e na Inglaterra, a normal troca de informação entre os vários países foi comprometida, e como consequência o projecto teve mais de 2,5 mil milhões de dólares de prejuízo e atrasos superiores a um ano (Rothman, 2006).

No entanto, e mesmo num ambiente industrial normalizado, a troca de informação necessita ser efectuada de uma forma profissional e baseada em modelos semânticos (i.e. contextualizando a informação trocada), o que permite quebrar a distância entre os processos e ferramentas heterogéneas utilizadas pelas diferentes organizações, bem como colmatar as diferenças geográficas, organizacionais e culturais existentes.

Estas questões da troca de informação têm vindo a ser retratadas e estudadas no domínio referido como a Interoperabilidade. O IEEE define interoperabilidade como “a capacidade de dois ou mais sistemas ou componentes trocarem informação e poderem utilizar a informação trocada” (IEEE, 1990). Ou seja, a simples troca de informação não é suficiente e nenhuma mais-valia é acrescentada ao ciclo de vida caso a informação trocada não possa ser utilizada adequadamente. É imprescindível que os interlocutores, além de compreenderem a forma como a informação é descrita, a possam interpretar e utilizar correctamente (i.e. depreenderem a semântica associada).

Por outro lado, o facto de se trocar a informação de forma interoperável (i.e. a informação poder ser utilizada), não garante a fácil interpretação da mesma. As trocas, principalmente dos modelos CAD, tendem a conter uma grande quantidade de dados, o que pode conduzir a uma falta de objectividade na interpretação da informação. A troca de dados entre os intervenientes de diferentes especialidades pode trazer também problemas, já que a

---

<sup>1</sup> Airbus© A380 - <http://www.airbus.com/en/aircraftfamilies/a380/>

<sup>2</sup> CATIA - Computer Aided Three-Dimensional Interactive Application - <http://www.3ds.com/products/catia/>

---

informação que é trivial num domínio, pode não o ser em outros, levando a uma dificuldade acrescida na interpretação da informação. Por outro lado, no caso de os sistemas trabalharem com grandes quantidades de informação torna-se um problema, pela necessidade de maiores e melhores recursos cujos custos são de considerável valor.

Para colmatar a dificuldade de interpretação da informação e melhorar a percepção da mesma, é utilizada outra técnica, a Visualização, que define a forma de melhorar o entendimento da informação por quem a necessita recorrendo a formas intuitivas de representação (Wijk, 2006). Através de soluções avançadas de visualização permite-se melhorar a compreensão dos dados de produto ao longo de todo o ciclo de vida.

### 1.1. MOTIVAÇÃO

Para se melhorar o desempenho dos intervenientes durante o ciclo de vida do produto, a informação deve ser trocada eficientemente e poder ser utilizada de forma apropriada no realizar das funções específicas de cada interveniente. No entanto, a multidisciplinaridade e densidade do seu conteúdo implicam uma necessidade de extrapolação da informação, tendo em vista uma melhor percepção orientada às necessidades específicas dos intervenientes. Por exemplo, os gestores não querem dissecar a informação técnica referente à concepção do produto, querem ter acesso imediato a informações relevantes ao negócio, como os materiais necessários, os custos dos mesmos, as horas de trabalho, os intervenientes nos processos, etc.

Através da técnica da visualização é possível extrapolar (i.e. processar e agregar a informação) a informação do modelo, de forma a representar a especificidade necessária a cada interveniente no seu próprio domínio. A extrapolação passa por vários estágios partindo do modelo de informação até chegar à forma de mostrar a informação da maneira mais adequada para a realização da função pelo interveniente, primeiro filtrando a informação relevante ao domínio e seguidamente trabalhando-a fazendo algum processamento que enriqueça a sua utilização. Assim, e através de técnicas de visualização é possível melhorar a percepção da informação e, assim, incrementar o desempenho das funções próprias de cada interveniente durante o ciclo de vida do produto.

Para mais, e no caso específico da utilização dos sistemas CAD para a criação dos modelos de informação utilizados no ciclo de vida dos produtos, é importante que todos possam ter

---

acesso à informação de forma adequada, mesmo sem terem acesso aos sistemas CAD. Mas a utilização e interpretação da informação descrita pelos modelos criados nos sistemas CAD não é trivial para todos, sendo necessário recorrer a formação avançada dos intervenientes para aprender a lidar com a complexidade e utilização dos sistemas CAD. Esta solução implica tanto custos elevados bem como uma necessidade extra de tempo, promovendo o contrário do desejado, isto é, uma menor eficácia e eficiência no ciclo de vida do produto.

Por outro lado, ao se recorrer a diversas entidades com diferentes competências no ciclo de vida do produto, nem sempre se consegue garantir que as trocas de informação aconteçam com sucesso. Quanto mais entidades (externas ou internas) estiverem envolvidas no desenvolvimento, menor é a probabilidade de todas estas utilizarem os mesmos sistemas, levantando-se assim possíveis incompatibilidades entre formatos de troca de dados. Esta situação de não-interoperabilidade conduz a uma dificuldade acrescida na troca e utilização da informação com o prejuízo do desempenho e eficácia da colaboração ao longo do ciclo de vida do produto.

Mesmo com a troca bem sucedida de informação e com a possibilidade da mesma poder ser interpretada correctamente, como se está perante um ambiente de trabalho colaborativo, a distribuição da informação pelas várias entidades conduz a questões de protecção da propriedade intelectual. Com a inclusão do trabalho colaborativo nos moldes atrás referidos, cria-se uma ligação com entidades externas às empresas, distribuindo-se a informação de desenvolvimento dos produtos a estas entidades. Essas trocas podem conter informação sensível para o negócio que não devem ser partilhadas a terceiros, como por exemplo, na troca de informação geométrica com o fim de se criar uma embalagem para o produto, filtrando-se os dados, de forma a ser trocado apenas uma representação dos limites do produto e não do seu conteúdo de engenharia, muitas vezes reduzindo-se a precisão do modelo (i.e. retirando-se detalhes, diminuindo-se resolução dos modelos, etc.).

Também, a integração de todos estes sistemas num ambiente de desenvolvimento colaborativo é uma necessidade, tal como a extensão dos já existentes (sistemas legados) com a inclusão de novas funcionalidades. Mas esta integração, devido à existência de inúmeras barreiras tecnológicas (formas de desenvolvimentos, modelos de dados, etc.), é dificultada o que implica desintegração ao longo de todo o ciclo de vida do produto.

---

Em suma, do ponto de vista industrial é necessário criar uma solução que permita resolver estas questões e que ao mesmo tempo seja robusta do ponto de vista de sistemas para simplificar a sua integração no ambiente já existente. Esta solução permitiria resolver problemas como a dificuldade de aquisição de dados previamente utilizados, reduzir os excessos de custos e atrasos de prazos, aumentar a reacção a alterações de requisitos e ao avanço das tecnologias, não se obter o desempenho previsto e a falta de suporte aos produtos existentes e sistemas.

Já existe um esforço de referência neste sentido, denominado Organizações suportadas em Modelos, do inglês, *Model Based Enterprise* (MBE) (PDES, Inc., 2007). Este agrupa três áreas, engenharia baseada em modelos, manufactura baseada em modelos e sustentação baseada em modelos. O MBE centra-se no desenvolvimento orientado pelos modelos aplicado a todas as áreas do desenvolvimento dos produtos e conta como características como a utilização de modelos para definir a informação de negócio de uma forma abstracta para poder ser aplicada em qualquer domínio onde seja necessária, as aplicações e processos são integrados recorrendo-se a informação normalizada, a informação é acessível através de interfaces para o acesso a dados, a possibilidade de se testar os modelos através de métodos matemáticos e simulações baseadas em métodos científicos, e a captura, troca e respostas de conhecimento dentro do ciclo de vida do produto.

## 1.2. CONTEXTO

A criação de soluções robustas para as questões identificadas – a dificuldade de troca de informação pela existência de vários formatos, a protecção da propriedade intelectual, e a apresentação da informação de uma forma adequada às necessidades dos intervenientes – passa pela adopção de práticas de referência na indústria, como por exemplo o MBE. O MBE assenta no paradigma do desenvolvimento orientado pelos modelos (MDD, do inglês, *Model Driven Development*), que consiste fundamentalmente no desenvolvimento de soluções de uma forma abstracta e independente da plataforma tecnológica, através da modelação dos elementos representativos da organização, e suas relações (e.g. ligação entre processos, produtos, organização, sistemas, etc.). Estes modelos são utilizados ao longo de todo o ciclo de vida do produto, desde a concepção, à engenharia e à sustentação (i.e. fase aquando o produto já se encontra no mercado, incluindo por exemplo suporte técnico ao mesmo), sendo utilizados para a interligação das várias fases. Este paradigma de desenvolvimento

---

permite solucionar os problemas de interoperabilidade dentro do ciclo de vida do produto, já que passa a existir uma descrição explícita e normalizada do conhecimento associado ao produto.

O desenvolvimento orientado pelos modelos centra-se na aplicação generalizada de modelos, onde o seu papel passa pela descrição de todos os elementos do sistema e suas relações, regendo-se por um princípio de “tudo é um modelo”, i.e. todas as componentes do sistema são descritas através de modelos. Neste contexto, de utilização abrangente de modelo, é necessário interoperar os diferentes modelos do sistema, e portanto as operações entre modelos (e.g. transformação de modelos) tornam-se um factor essencial para a aplicação de uma solução de desenvolvimento suportada por modelos.

Com a descrição de todos os formatos existentes para a troca de dados geométricos em modelos, podem-se aplicar transformações entre eles, transformando a informação descrita por um formato numa equivalente descrita em um outro. Também se aplicam transformações para extrapolar a informação apropriada para ser apresentada de forma intuitiva, e.g. de uma forma visual, para o trabalho a se realizar. Por último, as transformações de modelos são também utilizadas na contextualização da informação representada, de forma a se poder, por exemplo, proteger o seu conteúdo nas trocas com terceiros, aplicando-se as regras de protecção de propriedade intelectual à operação.

Neste trabalho, estas transformações surgem aplicadas ao domínio da visualização, contribuindo para a realização de um outro conceito emergente, onde se aplica o paradigma do desenvolvimento orientado pelos modelos à visualização, resultando na visualização orientada pelos modelos. Esta consiste, por analogia com o desenvolvimento orientado pelos modelos, realizar as vistas definidas pela visualização através do uso de modelos.

Esta dissertação tem como objectivo a proposta de uma solução que assente no paradigma da visualização orientada pelos modelos, e que permita a integração de ambientes industriais colaborativos. Esta solução deve ser integrável nas ferramentas actualmente existentes de forma a poder ser utilizada sem a necessidade de se alterar a forma de trabalho dos profissionais envolvidos no ciclo de vida do produto.

---



### 1.3. METODOLOGIA DE INVESTIGAÇÃO

A metodologia de investigação adoptada neste trabalho baseia-se nos princípios do método científico, que envolve os seguintes passos gerais (Schafersman, 1994):

1. Definição da questão;
2. Colecção de informação e recursos;
3. Formação de uma hipótese;
4. Elaboração de uma experiência;
5. Análise de resultados e conclusões;
6. Publicação de resultados.

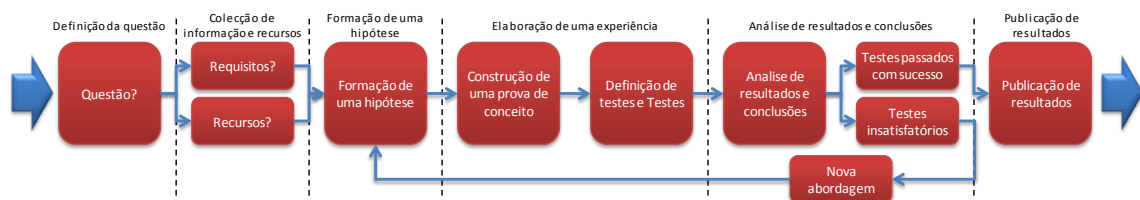


FIGURA 1.2 – METODOLOGIA DE INVESTIGAÇÃO.

#### 1. DEFINIÇÃO DA QUESTÃO

Um trabalho de investigação inicia-se com a definição de um problema. A questão desta dissertação advém do desenvolvimento de uma plataforma para a criação de catálogos de produtos, e consiste em permitir aos utilizadores, sem recorrerem a sistemas CAD, conseguirem visualizar o modelo tridimensional do produto e extraírem informação relevante, e.g. os materiais utilizados. Este trabalho enquadra-se também num projecto de investigação e desenvolvimento financiado pela comissão europeia no âmbito do sexto programa quadro, nº 31139, com o acrónimo INNOVAFUN<sup>3</sup>, no âmbito na indústria do mobiliário, para implementação de normas para a interoperabilidade.

#### 2. COLECÇÃO DE INFORMAÇÃO E RECURSOS

Os requisitos interpostos pela questão têm de ser identificados através de uma confluência das necessidades da indústria relativamente à visualização e dos desafios de investigação ambicionados. Como o foco do trabalho é na visualização, é necessário fazer a caracterização desta técnica no contexto do ciclo de vida do produto. Como existem diversas

<sup>3</sup> INNOVAFUN (INNOVate FURNiture) - <http://standards.eu-innova.org/Pages/InnovaFun/default.aspx>

formas de trocar dados de geometria, procede-se ao levantamento das mais relevantes no domínio, interpretando a sua ligação com os requisitos definidos. Por outro lado, para o desenvolvimento da solução é necessário fazer um levantamento dos recursos existentes, neste caso o paradigma da visualização orientada pelos modelos.

### **3. FORMAÇÃO DE UMA HIPÓTESE**

Com base nos requisitos definidos e dos recursos disponíveis, parte-se para uma realização conceptual que sirva como hipótese de investigação. Esta é a base da solução, e deve possibilitar tanto uma abordagem conceptual, ou seja, uma análise ao problema de um ponto de vista abstracto, como também a sua implementação, de forma a se poder partir para a elaboração de uma experiência.

### **4. ELABORAÇÃO DE UMA EXPERIÊNCIA**

A hipótese de investigação atrás descrita é sujeita a um processo de validação através de experimentação. Para tal realiza-se uma implementação sobre a forma de uma prova de conceito, para se poder efectuar testes para a validação da hipótese proposta em resposta à questão definida.

### **5. ANÁLISE DE RESULTADOS E CONCLUSÕES**

Para se aferir a solução proposta, define-se uma bateria de testes que permitam verificar se esta cumpre com os requisitos definidos. Estes testes são aplicados em ambiente controlado através de experimentação para validar a solução. Caso os testes falhem, volta-se ao ponto 3, e procede-se à refinação da solução do problema, até se chegar a uma nova solução que responda com sucesso aos requisitos definidos. No final estabelecem-se as conclusões com base nos resultados obtidos na análise.

### **6. PUBLICAÇÃO DE RESULTADOS**

Como resultado deste trabalho foram publicados três artigos científicos:

- Almeida, B.; Agostinho, C.; Nunez-Arino, M.J.; Jardim-Goncalves, R., "Model morphisms as an enabler for open visualization of product data," Intelligent Systems, 2008. IS '08. 4th International IEEE (Institute of Electrical and Electronics Engineers) Conference , vol.1, no., pp.8-29-8-34, 6-8 Sept. 2008.
-

- C. Agostinho, B. Almeida, M.J. Nuñez-Ariño, R. Jardim-Gonçalves, "Interoperability and Standards: The way for Innovative Design in Networked Working Environments", CIRP (College International pour la Recherche en Productique) Design Conference 2009, Cranfield University, Cranfield, UK, March 2009.
- Bruno Almeida, Pedro Maló, Marco Delgado, Ricardo Jardim-Gonçalves, "Integração de ambientes industriais colaborativos através da visualização orientada pelos modelos", AISTI (Associação Ibérica de Sistemas e Tecnologias de Informação) 4ª CISTI (Conferência Ibérica de Sistemas e Tecnologias de Informação) 2009, Póvoa de Varzim, Junho 2009.

O trabalho constou também num dos entregáveis do projecto INNOVAFUN, mais concretamente o D2.1, "Services for funStep standard adoption, and design of new business practices".

#### 1.4. ESTRUTURA DE CAPÍTULOS

Este documento é composto por sete capítulos, que são:

1. Introdução: Neste capítulo, são apresentados, o enquadramento e objectivos desta dissertação, bem como a metodologia de investigação adoptada.
  2. Visualização de dados de produto em ambientes industriais: Este capítulo é composto por duas partes. Na primeira, são levantados os requisitos e os impactos da visualização num ambiente de trabalho industrial. Numa segunda, são listados os formatos actualmente utilizados na troca de dados geométricos, onde é incluída uma pequena análise da aptidão dos mesmos no contexto da visualização.
  3. Visualização suportada em modelos: Iniciado com uma explicação do que consiste desenvolvimento orientado pelos modelos, acaba com o levantamento de como os mesmos são actualmente aplicados na área da visualização.
  4. Solução interoperável para a visualização de dados de produto: Aqui começa-se com uma definição mais objectiva dos requisitos para se definirem os fundamentos que vão suportar a solução. Esse caminho é depois completado, com a criação de um da solução, que acaba numa arquitectura orientada pelos modelos.
  5. Implementação da solução: Este capítulo descreve os passos efectuados na criação da prova de conceito. Começando pela escolha das tecnologias a se instanciarem à
-

arquitetura, passando depois pelo processo de implementação e acabando na criação de uma interface para a sua utilização.

6. Testes e validação: Este capítulo começa pela definição de quais os testes adequados para se testar a conformidade da solução proposta com os requisitos definidos. De seguida, passa-se para a aplicação desses testes à prova de conceito, acabando o capítulo numa análise dos resultados obtidos nos testes de forma a avaliar o nível de desempenho da solução.
  7. Conclusões e perspectivas futuras: O último capítulo desta dissertação apresenta a conclusão do trabalho efectuado. Essa análise passa por pontos como o impacto do mesmo se adoptado a um nível industrial e quais os pontos que podem ser evoluídos, quer do ponto de vista técnico, como o científico.
-

## 2. VISUALIZAÇÃO DE DADOS DE PRODUTO EM AMBIENTES INDUSTRIAIS

Devido à abundância de funcionalidades nas novas tecnologias, o desenvolvimento de produtos centra-se, hoje em dia, num ciclo de desenvolvimento assistido por computador, com o objectivo de reduzir os custos do desenvolvimento e obter um menor tempo de colocação dos produtos no mercado. Para isso, recorre-se à criação de protótipos, à execução de simulações e, até mesmo, permitir, através do uso de realidade virtual, testar a utilização dos produtos numa simulação do seu ambiente real.

A visualização é um meio essencial no ciclo de desenvolvimento assistido por computador. Esta melhora a partilha da informação entre profissionais, e entre sistemas, com o objectivo de realizar um produto mais capaz, mais depressa e com menores custos. A visualização é relevante e apresenta mais-valias ao longo do todo o ciclo de vida do produto, que, segundo a PDMA (King, 2007), é dividido em três grandes fases: Descoberta, Desenvolvimento e Comercialização.

A fase de **Descoberta** tem o seu início aquando da existência de uma nova ideia, e consiste na definição de requisitos e na criação de desenhos preliminares. Nesta fase, na criação do desenho conceptual do produto, surge a primeira necessidade de utilização de ferramentas de CAD. Aqui as técnicas de visualização começam a tornar-se uma mais-valia, suportando desde logo, a observação e validação preliminar do conceito (forma, função, adequação, estética, etc.) e, também, permitindo a outros agentes envolvidos (profissionais, clientes, etc.) que interagem no desenvolvimento da ideia e conceito associado, e não são utilizadores dos sistemas CAD, acompanharem e intervirem no desenrolar da concepção do produto.

O **Desenvolvimento**, onde existe uma forte utilização de ferramentas CAD, passa pelo desenho detalhado do produto, definindo-se as questões de engenharia, as partes constituintes, etc. Com o passar dos anos e com a utilização de sistemas CAD no desenvolvimento dos produtos, as organizações formaram um vasto catálogo digital de produtos previamente desenvolvidos, levando à possibilidade do reaproveitamento de componentes já utilizadas noutros produtos, espelhando muitas vezes soluções já testadas e com provas dadas. É nesta fase que se recorrem às técnicas fornecidas pela engenharia assistida por computador (CAE) e à prototipagem rápida, reduzindo-se o tempo de execução

---

desta fase, utilizando-se a criação rápida de protótipos, análises aos modelos CAD (e.g. análises de stresse, térmicas, dinâmica de fluidos, etc.), a utilização de realidade virtual para a simulação dos produtos, etc. A visualização, nesta fase, permite observar, inspeccionar e testar diferentes soluções, conduzindo, assim, por exemplo, à antecipação de erros de desenvolvimento, tanto técnicos como conceptuais, recorrendo-se à promoção da circulação de informação entre os intervenientes, incluindo os próprios clientes, que com o acesso ao estado do desenvolvimento dos produtos podem promover alterações que vão ao encontro das suas necessidades ou ambições.

A última destas três fases, a **Comercialização**, tem a contribuição da manufactura assistida por computador (CAM) no auxílio aos métodos de produção, criando-se, se necessário, moldes para as ferramentas utilizadas na manufactura, gerando-se instruções de comando para as máquinas de produção, etc. Aquando da produção em larga escala, nesta fase podem ser efectuadas simulações das linhas de produção, de forma a se otimizar o processo produtivo. Por fim, e para garantir a qualidade dos produtos, são definidos métodos de análise para se aferir a adequada especificação do produto. Aqui está também presente o trabalho de venda e manutenção do produto, como o *marketing* e a assistência técnica. Nesta fase, a visualização é importante na simulação, monitorização da linha de produção e na formação dos operadores. Esta é também fundamental para a criação de manuais de utilização e manuais técnicos (podendo-se recorrer a interfaces avançadas, animações, etc.) e à exploração dos modelos do produto em acções de marketing e venda.

A visualização é, assim, um elemento central no ciclo de vida do produto, permitindo agregar os intervenientes no desenvolvimento do produto, integrar as fases de desenvolvimento (desenho, manufactura, negócio, venda, etc.) através da troca de informação estritamente necessária e suficiente ao desempenhar de cada função. A visualização contribui, assim, para um aumento de produtividade, uma maior criatividade e melhores e mais capazes produtos, que se traduz em menores tempos de desenvolvimento e menores custos ao longo de todo o ciclo de vida do produto.

## 2.1. VISUALIZAÇÃO NO CICLO DE VIDA DO PRODUTO

Nas várias fases de desenvolvimento de um produto, a informação é um bem valioso, por isso, deve ser disponibilizada a todos os intervenientes, independentemente dos processos e ferramentas que cada um utiliza, bem como dos seus domínios de actividade. Como a

---

informação envolvida no desenvolvimento dos produtos abrange um elevado número de parâmetros e aplicações, torna-se de difícil interpretação por parte dos profissionais, sendo que as soluções de visualização, de modo interactivo e gráfico, possibilitam um melhor acesso à mesma, facilitando o trabalho aos vários intervenientes permitindo a utilização de informação dos mais diversificados domínios que se regem por ferramentas e processos distintos sem se ter de alterar os seus próprios processos (MacKrell, *The Role of Product Data in Creating Innovation*, 2006).

### 2.1.1. IMPACTOS DA VISUALIZAÇÃO

A visualização, dentro do ciclo de vida do produto, pretende ajudar nas dificuldades de interpretação da informação devido a esta nem sempre ser do domínio de utilização, ser de grande complexidade ou quantidade, e também existir em ambientes heterogéneos, tanto ao nível de sistemas, como de formas de trabalho e de troca de informação. Assim e para se desenvolver uma solução que ajude nestas dificuldades é necessário proceder ao levantamento de quais os impactos e quais as áreas em que a visualização é relevante.

Esta identificação é efectuada nas três grandes fases gerais definidas anteriormente (Descoberta, Desenvolvimento e Comercialização), mas subdividindo-as em quatro actividades mais específicas (Concepção e engenharia<sup>1</sup>, Suporte à cadeia de fornecimento<sup>2</sup>, Planeamento de manufactura e produção<sup>3</sup> e Suporte ao mercado<sup>4</sup>), relacionando-se como mostra a figura seguinte.

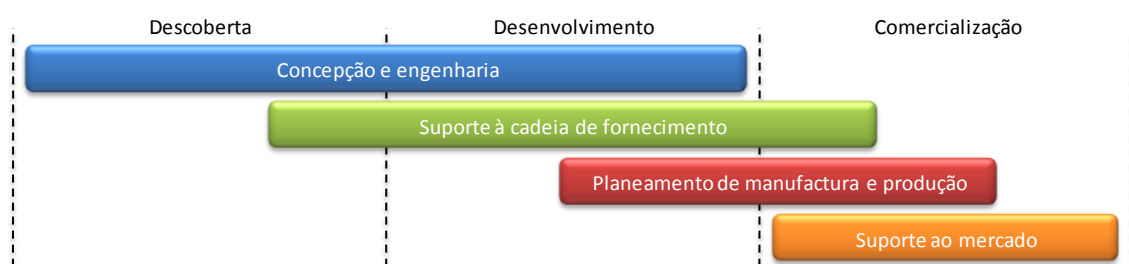


FIGURA 2.1 – CICLO DE VIDA DO PRODUTO VS ACTIVIDADES ESTUDADAS NO IMPACTO DA VISUALIZAÇÃO.

#### 2.1.1.1. CONCEPÇÃO E ENGENHARIA

Os projectistas querem reutilizar os dados de produtos já desenvolvidos, ao invés de os recriar, perdendo tempo e dinheiro já dispendido anteriormente. Utilizando a visualização, é dada a possibilidade de os gestores de projecto analisarem o trabalho já efectuado para identificarem o que pode ser reaproveitado. Por outro lado, nos produtos a serem

desenvolvidos é dada a possibilidade de identificarem as várias partes, componentes e materiais necessários, permitindo decidir quais as que necessitam de novo desenvolvimento e as que podem ser efectuadas com o reaproveitamento de outras partes já desenvolvidas.

Ainda, os erros de especificação e de concepção que não sejam corrigidos a tempo, podem provocar problemas sérios ao desenvolvimento do produto em fases mais avançadas. Com a visualização pretende-se que um variado número de intervenientes, desde os gestores até aos clientes, tenham acesso antecipado ao desenho, e possam avaliar as alterações de forma a se diminuir os erros de desenvolvimento e a promover alterações mais rápidas para ir de encontro às necessidades e objectivos do produto.

Um produto, sendo, por norma, a junção de várias componentes, necessita ser avaliado como um todo, mas nem sempre estas são desenvolvidas recorrendo-se às mesmas ferramentas. Com a visualização, possibilita-se a junção das várias componentes, independentemente do seu formato de troca, permitindo analisar se elas coexistem como previsto.

A criação de um produto segue um objectivo, uma utilização no mundo real. A visualização, possibilitando a simulação desse produto num ambiente real, permite, que os engenheiros analisem se este corresponde às especificações. Uma mais-valia nesta fase, é a utilização de técnicas de realidade virtual, para se poder entrar virtualmente no ambiente simulado.

#### 2.1.1.2. SUPORTE À CADEIA DE FORNECIMENTO

Com os fornecedores a contarem com vários clientes, e nem todos a utilizarem as mesmas ferramentas, cria-se uma barreira à troca de informação. Em analogia à junção das várias componentes dos produtos, também aqui as técnicas de visualização permitem harmonizar os diversos modelos representados por diversos formatos, permitindo aos clientes o acesso aos modelos trocados.

A inclusão de mecanismos de marcação na troca de informação permite que se troquem opiniões directamente nos modelos. Com a associação das opiniões às partes específicas dos modelos, quem interage com elas pode-se focar-se apenas nos comentários que lhe são relevantes e não em todos os comentários presentes no modelo global. Com a visualização, pode-se apresentar métodos, para a leitura, escrita e revisão desses comentários.

---



Também, com a inclusão de terceiros no desenvolvimento de produtos, é necessário proteger a propriedade intelectual. Recorrendo-se à visualização pode-se definir vistas distintas sobre os dados, trocando-se apenas a informação essencial, protegendo-se informação sensível para o negócio, sem prejudicar o desempenho do trabalho.

#### 2.1.1.3. PLANEAMENTO DE MANUFATURA E PRODUÇÃO

Saber de antemão qual o produto a produzir, permite analisar a capacidade do equipamento que irá ser utilizado na produção, diminuindo-se o tempo de produção e a probabilidade de acontecerem erros (i.e. a máquina não conseguir produzir a peça desejada). Com as técnicas de visualização, pode-se analisar e inclusive simular o desempenho do equipamento.

Encontrar formas de diminuir o tempo de entrada dos produtos no mercado, aumenta a sua competitividade. Recorrendo-se à visualização, possibilita-se aos profissionais que procuram fornecedores de peças que entendam, numa fase inicial, as componentes constituintes do produto, permitindo uma pesquisa antecipada desses fornecedores, diminuindo o tempo de entrada de mercado dos produtos.

Os operadores de máquinas tendo disponível uma visualização da peça a produzir conseguem verificar mais rapidamente a eficiência da linha de montagem (i.e. verificar se as peças produzidas correspondem às desenhadas). Também na formação dos profissionais que montam as diversas peças, é possível usufruir das vantagens da visualização, com a utilização de animações demonstrativas de como o seu trabalho deve ser efectuado.

Existem ainda indústrias que contam com elevados padrões de regulamentação, onde muitos dos seus processos são definidos por autoridades reguladoras. Por exemplo, produtos que tenham embalagens normalizadas, podem utilizar a visualização para extrair a meta-informação (i.e. informação como o nome do produto, seus constituintes, etc.) dos modelos e incluí-la automaticamente no processo da criação de embalagens, baixando ou mesmo anulando a interacção humana, reduzindo-se os tempos de produção.

#### 2.1.1.4. SUPORTE AO MERCADO

A criação de documentação associada ao produto é uma tarefa relevante na sua comercialização. Utilizando a visualização é possível a criação de ilustrações a partir dos modelos originais, evitando à recriação das mesmas. Também se pode, por exemplo, incluir

---

animações (para facilitar o uso dos produtos) ou instruções técnicas, em manuais interactivos (e.g. para suporte e manutenção).

Com o emergir do comércio electrónico, a criação de catálogos necessita assentar sobre uma estrutura que possibilite o acesso aos conteúdos sem barreiras (i.e. sem se recorrer a ferramentas específicas, e.g. utilizando apenas navegadores Web). A visualização permite então criar esta estrutura comum, harmonizando as representações dos produtos descritas nos mais diversos sistemas, reaproveitando os desenhos existentes para uma representação mais perto da realidade dos produtos.

Produtos que necessitem de assistência no local do cliente, nem sempre contam com técnicos das próprias marcas a providenciar a assistência, nem com profissionais devidamente formados para a maioria dos problemas. Logo acesso a informação de reparação diminui o tempo de resposta das equipas. A utilização de repositórios onde exista informação de reparação, com detalhes técnicos, animação, etc. é uma mais-valia, e com a possibilidade dos técnicos actualizarem essa informação com a sua experiencia permite que outros técnicos consigam responder a problemas idênticos mais rapidamente, conduzindo a um suporte mais eficiente e uma maior satisfação dos clientes.

Como já foi referido anteriormente, permitir às áreas de marketing presentes no ciclo de vida do produto a possibilidade de reaproveitarem o trabalho de engenharia já efectuado, é uma forma de melhorar a apresentação dos produtos a eventuais compradores, bem como poupar tempo na entrada do mesmo no mercado. Criar publicidade que contenha uma forte componente visual do produto, ajuda à escolha do mesmo em detrimento de outros, apresentados de forma textual que não chamam à atenção dos potenciais compradores (MacKrell, *Extracting Value from Your Product and Process Data*, 2006).

#### 2.1.2. DESAFIOS PARA A APLICAÇÃO DA VISUALIZAÇÃO

Apresentado o impacto que a adopção da visualização pode ter ao longo de todo o ciclo de vida do produto, procede-se ao levantamento de quais os grandes desafios que a implementação com sucesso do conceito da visualização levanta.

Ao nível da troca de dados que permite aplicar a visualização, é imperativo existir um ambiente interoperável. A criação deste ambiente, passa pela adopção de modelos de troca de dados de referência, facilitando a integração do ciclo de vida do produto, i.e. que possa

---

ser utilizado pelas diversas ferramentas, sendo portanto utilizado em sistemas CAD, CAE, CAM, de colaboração, de gestão, visualizadores, publicações técnicas e outros. Facilitando o acesso à informação pelos projectistas, engenheiros, vendedores, compradores, equipas de marketing, administradores, gestores e a muitos outros cuja informação possa ser útil desde a ideia e concepção do produto até este ser extinto do mercado.

Logo, como grande parte dos produtos colocados no mercado são desenvolvidos recorrendo aos mais diversificados sistemas, estes são guardados nos mais variados formatos e versões, logo, um dos desafios é: se será possível construir uma base unificadora de modo a aplicar a visualização em ambientes industriais onde existe heterogeneidade de formatos e sistemas?

Visto os profissionais utilizarem a informação consoante o seu domínio de trabalho, esta é descrita de várias formas adaptadas ao contexto de utilização, e.g. desenhos técnicos, fotografias, modelos tridimensionais, listagens de partes, informação de produção, custos, instruções para o manuseamento, e os mais diversificados dados de produto. Recorrendo-se portanto às técnicas de visualização pode-se apresentar a informação adaptada ao domínio de aplicação de uma forma simples e de fácil interpretação, e.g. recorrendo-se a facilitadores visuais orientados ao domínio de trabalho específico do interveniente

Portanto, como inúmeros intervenientes no processo são de diferentes domínios, nem todos necessitam do mesmo tipo de informação para efectuar o seu trabalho, será possível a preparação da informação de forma a conter apenas a necessária para o seu destinatário e dar a possibilidade de esta ser apresentada de uma forma visual intuitiva?

Levar aos profissionais que tomam decisões, informações precisas sobre o desenvolvimento, ajuda-os a tomarem as decisões acertadas. Um dos problemas que a visualização propõe colmatar, é a necessidade de se precisar de diversas ferramentas para o acesso à informação relevante para o desempenho das funções dos intervenientes no ciclo de vida do produto.

Então, alterar as ferramentas de trabalho utilizadas provoca um revés na sua eficiência, e um aumento dos custos associados, quer nas licenças, quer na formação, será portanto possível integrar o modelo unificado nas ferramentas de trabalho já existentes?

Será então possível conceber uma solução interoperável, capaz de criar uma tal base unificadora para a visualização de dados de produto, que se integre nas soluções já

---

existentes e permita uma personalização para contextos de aplicação específicos? Este é o desafio a que se propõe esta dissertação!

## 2.2. NORMAS DE TROCA DE DADOS PARA A VISUALIZAÇÃO

Visto as organizações utilizarem soluções para a visualização de dados de produto em poucos processos, é necessário promover a sua utilização para se levar a informação a áreas antes inexploradas. Os sistemas CAD são uma forte base de desenvolvimento de produtos, e estão presentes ao longo de todo o seu ciclo de vida, o que leva a que actualmente apenas exista colaboração entre quem lhes tem acesso e está apto para os utilizar, limitando o número de profissionais que podem participar activamente em todas as actividades do ciclo de vida do produto. Logo, é imperativa a existência de uma forma de troca de informação entre os vários intervenientes, para posteriormente se aplicar as técnicas de visualização.

Uma das formas de suportar esta interoperabilidade é a aceitação de formatos de troca de referência como suporte para a troca de dados na criação de ferramentas de que permitam a visualização e reutilização de toda a informação trocada. Hoje em dia já existem vários formatos, que contam com ferramentas desenvolvidas sobre eles e que permitem este tipo de utilização (e.g. interacção dinâmica com os conteúdos, como a rotação, ampliação, desagregação de partes, etc.).

### 2.2.1. CARACTERÍSTICAS RELEVANTES

Com diversos formatos existentes para a troca de dados, existem várias características que definem a sua aptidão para a visualização, que tanto podem ser técnicas como de negócio.

Ao nível técnico as considerações começam pelo tipo de informação que se pode descrever e trocar, variando entre informação CAD (e.g. partes constituintes, montagem, etc.), dados de visualização (e.g. a forma como o utilizador pode interagir com os objectos, animações do seu uso, etc.), descrição da informação geométrica (e.g. descrevendo os objectos em elementos geométricos básicos, como linhas, cubos, etc.) e chegando ao nível mais baixo, representando-se os objectos pela descrição das suas superfícies.

Com modelos complexos a poderem ter ficheiros na grandeza dos megabytes ou mesmo gigabytes, podem existir dificuldades acrescidas na troca de informação num ambiente colaborativo. Para se facilitar essa troca é necessário saber qual o objectivo de utilização dos dados. Por exemplo, para os criadores de embalagens de produto, os modelos não

---

necessitam da complexidade dos dados de engenharia e a sua precisão pode ser suficiente se descrita na ordem dos milímetros. Portanto, pode-se portanto jogar com a precisão do modelo para controlar o tamanho dos ficheiros. Por exemplo, um formato que descreva as superfícies geométricas através de polígonos (e.g. triângulos) conta com a sua precisão aumentada com a utilização de mais polígonos por unidade de medida, aumentando, por consequência, o tamanho dos ficheiros.

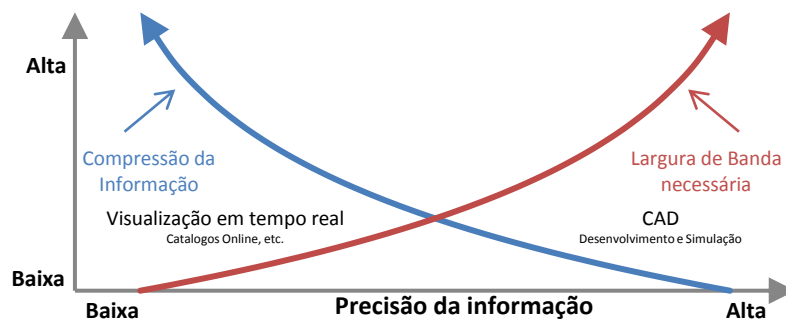


FIGURA 2.2 – RELAÇÃO ENTRE A LARGURA DE BANDA UTILIZADA E COMPRESSÃO DA INFORMAÇÃO FACE À PRECISÃO DA INFORMAÇÃO.

A utilização de diversas formas de troca de informação também pode melhorar a colaboração. Por exemplo a utilização de técnicas de *streaming*, trocando-se apenas a informação necessária no momento, minimiza o uso das comunicações. Quando a troca é realizada por formato físico (e.g. DVD), pode-se obter um grande nível de precisão sem custos adicionais, aproveitando-se todo o espaço disponível no meio de troca da informação.

Os formatos terem integração com as ferramentas actuais é também uma mais-valia. Alguns dos que irão ser analisados são inclusive os formatos padrão de alguns sistemas CAD, havendo outros, aceites globalmente, cujo essas ferramentas suportam a sua importação e exportação permitindo a fácil reutilização dos dados, outros são descritos em linguagens actuais (e.g. XML) e permitem uma integração facilitada nas ferramentas já existentes.

Ao nível do negócio entra em jogo um novo elemento, os custos. Cada organização tem as suas próprias necessidades, por isso terem acesso às técnicas de visualização a baixos custos torna-se um factor decisivo para a massificação da visualização. Como nem todas as organizações são iguais, para umas a visualização pode ser uma mais-valia, mas para outras, como aquelas em que os sistemas CAD já foram disseminados, implementar uma solução de visualização acarreta custos e pode não trazer grandes novidades ao panorama actual da empresa, visto que este conceito tem como objectivo permitir às empresas que não contam com estas soluções integradas, possam usufruir delas a baixos custos.

Quando se selecciona um formato para a visualização, também se está a escolher um novo parceiro de negócios, porque com a inclusão do mesmo nos processos das organizações estas ficam à mercê da sua eficácia. Uma escolha acertada passa por um formato que seja desenvolvido por entidades independentes que garantam que o mesmo seja evoluído com o tempo e que nenhuma das organizações envolvidas, na sua utilização, faça interpretações diferentes da sua especificação pondo em causa a interoperabilidade do mesmo. O caso fica favorável, quando os formatos são regulamentados por organizações internacionais de normalização, como a ISO ou a IEC, por consórcios industriais, como o W3C ou o IEEE, ou por agências nacionais de normalização, como a CEN ou a ANSI.

Apostar num formato que seja largamente e globalmente suportado, que garanta um constante desenvolvimento é uma mais-valia. Um formato que conte com inúmeras vantagens, mas não tenha a sua especificação pública, torna-o inadequado para o paradigma da visualização. Por isso, os formatos a serem utilizados devem contar com especificações públicas (serem formatos abertos). Mas o facto de serem públicas não implica serem gratuitas, por exemplo, o STEP, é uma norma pública mas tem um custo associado. Por isso a solução ideal para a visualização pode passar por um formato cuja especificação seja disponibilizada publicamente e de forma gratuita.

A existência de ferramentas que dêem suporte aos formatos é um factor de peso. Um factor que contribui para a aceitação de um formato é a quantidade e qualidade de ferramentas existentes que o consigam utilizar. Alguns formatos contam com comunidades de desenvolvimento que disponibiliza várias aplicações de forma gratuita, por vezes fornecendo apenas funcionalidades básicas, outras fornecendo funcionalidades mais avançadas, que dependendo das circunstâncias podem ser suficientes (e.g. visualizar os modelos).

Mesmo existindo a necessidade de recorrer a ferramentas pagas, ou a formatos que tenham custos associados, as vantagens que advêm da utilização da visualização, podem cobrir esses custos adicionais, aumentando a eficácia dos processos. Logo, fazendo o balanço final, os ganhos podem suplantar os custos, tornando-se um investimento.

A massificação da visualização depende em grande parte da aceitação dos formatos por parte das organizações, portanto formatos que tenham provas dadas e sejam utilizados por organizações de relevo, partem, logo à partida, com uma vantagem adicional em relação aos outros (MacKrell, *The Role of Product Data in Creating Innovation*, 2006).

---

### 2.2.2. NORMAS DE REFERÊNCIA

De seguida é apresentada uma breve descrição de alguns dos formatos utilizados actualmente na indústria, focando as suas vantagens e desvantagens para a utilização como um formato para a troca de informação para a visualização. Uma descrição mais pormenorizada de cada formato pode ser encontrada no ANEXO A (Página 103).

**3D XML** – Formato criado pela Dassault Systèmes em conjugação com a indústria com o objectivo de se trocar dados de forma rápida e eficiente. Baseado no XML, com a sua especificação pública e sem custos, este formato pode ser facilmente integrado nas ferramentas actuais, e permite tanto descrever uma grande quantidade de objectos geométricos, aliando alta precisão a baixos tempos de trocas, como conter informação de gestão dados trocados. Um ponto contra, é o formato ser criado e gerido por uma empresa de soluções CAD, dificultando a normalização do mesmo (Dassault Systèmes, 2005).

**ACIS – SAB (Standard ACIS Binary) e SAT (Standard ACIS Text)** – Mais que um formato, o ACIS é um dos núcleos de modelação existentes na indústria que alguns softwares CAD utilizam para a modelação de sólidos 3D. Este também descreve formatos para a troca de dados, que existem tanto na forma ASCII como binária, permitindo a análise do conteúdo sem visualizador próprio ou uma boa compressão de dados. A informação trocada é de modelação, resultando num elevado nível de precisão. Outra característica é a troca do histórico das operações, facilitando a reutilização dos dados. Com a evolução do formato e a inclusão de novas funcionalidades a especificação deixou de ser pública (Spatial Corp., 2007).

**DGN (Design)** – Formato proprietário da Bentley Systems, é utilizado como o formato padrão nas suas soluções. É um formato orientado para o CAD, descrevendo informação como as partes constituintes, as camadas de desenho, etc. contando com um elevado nível de precisão, que resulta em ficheiros de grandes dimensões. Um ponto a favor deste formato é a sua especificação ser pública e gratuita (Bentley Systems, Incorporated, 2008).

**DWF (Design Web Format)** – O DWF é um formato aberto, desenvolvido com o objectivo de uma distribuição eficiente e com protecção dos conteúdos da informação para quem necessite de ver, rever, ou imprimir os modelos. Sendo um formato orientado para a visualização os seus ficheiros são bastantes comprimidos (Autodesk, Inc., 2008).

---

**DWG (Drawing) e DXF (Drawing Exchange Format)** – Este formato descreve o mesmo tipo de informação que o DGN, sendo promovido por outra empresa, a Autodesk. Este tem duas versões, uma ASCII e outra binária, sendo que a binária evoluiu mais rapidamente por poder ser cifrada. Isto porque a especificação de este formato foi fechada, devido a rápida evolução do mesmo, mantendo-se a exclusividade das novas funcionalidades. Sendo este um dos formatos de referência na indústria, a Autodesk disponibiliza visualizadores gratuitos, bem como bibliotecas para a utilização do formato, mas com custos elevados. Para contrariar esses custos, uma aliança foi criada (Open Design Alliance) que providencia uma especificação gratuita, mas não completa, do formato (Autodesk, Incorporated, 2007).

**IGES (Initial Graphics Exchange Standard)** – Desenvolvido com o suporte do NIST foi criado com o intuito de promover um formato neutro para as trocas de modelos CAD, suportando toda a informação CAD. Apesar de contar com a sua especificação pública, várias empresas fizeram diferentes interpretações da mesma, levando à perda da interoperabilidade desejada. O formato é descrito em ASCII, mas a sua complexidade é tão elevada que os ficheiros normalmente são de dimensões proibitivas para a troca (NIST, 2002).

**JT** – Criado de raiz com o objectivo de ser utilizado na visualização este formato é muito leve, permitindo que seja transmitido de forma eficiente (e.g. utilizando-se métodos de *streaming*). A sua mais-valia está na descrição que este permite, e.g. parâmetros CAD associados aos modelos e parâmetros de visualização (e.g. a descrição da forma como os objectos interagem com luzes). Tendo por detrás a Siemens PLM software, não existe a garantia que, de futuro, o formato continue aberto sem restrições (UGS Corp., 2008).

**Parasolid XT** – Sendo parte de um núcleo de modelação tal como o ACIS, este formato existe para se poder efectuar as trocas dos produtos modelados dessa forma. Como este formato descreve a modulação dos objectos, a informação que consta nas trocas é essa mesmo, a de como o objecto foi modelado. Apesar de não ser regulamentado por uma entidade independente este formato conta com a sua especificação publicada o que promove a fácil integração nas ferramentas existentes (UGS Corporation, 2008).

**PDF (Portable Document Format)** – Um dos formatos mais utilizados para a troca de documentos na internet, com mais de um bilião de documentos existentes, conta agora com suporte para descrever modelos 3D. Uma das potencialidades que o destaca dos outros formatos, é o facto de se poder integrar animações 3D no meio de documentos, permitindo

---



o controlo pelo utilizador, sendo possível criar manuais interactivos, documentos técnicos com explicações pormenorizadas, etc. Sendo uma norma ISO de-facto, e a sua especificação ser disponibilizada gratuitamente, a garantia de evolução do formato é uma certeza, mas ao se incluir dados 3D em documentos, perde-se a possibilidade de reutilização dos mesmos (Adobe Systems Incorporated, 2008).

**SLC (Stereo Lithography Contour)** – Diferenciado dos outros por não ser na realidade um formato de representação 3D, este consiste em seccionar os objectos em várias fatias, e guardar a sua representação 2D, sendo o objecto 3D criado com o pós-processamento dessa informação. Uma das aplicações que este tipo de formato tem é, por exemplo na medicina, para guardar a informação proveniente dos TACs (Tomografia axial computadorizada), tirando imagens do corpo humano a diferentes profundidades (Kai, Jacob, & Mei, 1997).

**STEP (Standard for the Exchange of Product Data)** – Norma ISO 10303, não define somente um formato para a troca de informação, define toda uma norma para a representação de dados de produtos. Uma das suas componentes, a Parte 42, define como descrever objectos 3D, permitindo a representação de geometria complexa associada aos dados dos produtos. Apesar de poderosa a linguagem utilizada para essa descrição, é bastante complexa, o que dificulta a sua implementação. Ainda, não obstante se estar perante uma norma internacional, a mesma é paga, levando a que seja impeditiva para um desenvolvimento massificado de baixo custo (ISO - International Organization for Standardization, 2000).

**STL (Structured Triangular Language)** – Este formato representa os limites dos objectos, descrevendo-os através de triângulos. O formato em si é bastante simples, apenas são definidos os triângulos que realizam a fronteira do objecto. É possível transmitir os dados na forma binária ou ASCII, sendo possível um bom nível de compressão. A precisão depende directamente do número de triângulos a utilizar-se, sendo possível abdicar de alguma precisão para diminuir o tamanho dos ficheiros. Este formato é de momento dos mais utilizados no domínio da prototipagem rápida (Wu & Cheung, 2006).

**U3D (Universal 3D)** – Este formato foi criado com o intuito de permitir a transmissão de informação 3D via internet. O formato pode trocar informação CAD, possibilitando a reutilização da mesma em outros sistemas, mas também pode trocar informação orientada à visualização. A especificação é distribuída publicamente, e regulamentado pela ECMA, uma entidade independente, e conta com funcionalidades como a descrição de uma arquitectura

---

que permite a modificação da geometria em tempo de execução, um nível de detalhe contínuo associado a métodos de *streaming* (a informação é aumentada de qualidade aquando se aproxima) e permite também a definição de animações (ECMA, 2007).

**X3D (Extensible 3D)** – O X3D é uma norma ISO para a representação e troca de cenas 3D, sendo uma evolução de um dos primeiros formatos para o uso através da internet, o VRML. A sua especificação é aberta e o formato tem como um dos suportes físicos o XML. Visto ser um formato criado com o intuito da visualização, é possível definir animações, iluminação, a forma como se interage com os objectos e toda uma panóplia de parâmetros associados à visualização. Nas suas versões mais recentes o formato possibilita descrever informação CAD e meta-informação (i.e. relacionada com a gestão do produto) (Web3D Consortium).

### 2.2.3. ANÁLISE COMPARATIVA

De seguida é apresentada uma tabela com o resumo das características dos diversos formatos no que diz respeito à visualização, possibilitando a posterior análise comparativa dos mesmos. As características consideradas de relevo, resultam da identificação das características dadas como relevantes para as normas da troca de dados para a visualização e são agrupadas nos seguintes parâmetros:

- 1) Especificação:** Avalia se a especificação é aberta (i.e. disponibilizada publicamente de forma oficial pelos seus detentores) e se a mesma é distribuída de forma gratuita.
  - 2) Informação:** Qual o tipo de informação que o formato descreve. Se essa informação é uma descrição da geometria do objecto, utilizando a decomposição em objectos tridimensionais básicos, uma representação das suas superfícies, se conta com informação CAD, ou se tem suporte à troca de informação de visualização.
  - 3) Normalização:** Se o formato tem uma entidade a regulamentá-lo, tal como organizações internacionais (e.g. ISO e IEC), consórcios industriais (e.g. W3C e IEEE), ou entidades de regulamentação nacionais ou regionais (e.g. CEN e ANSI).
-

TABELA 2.1 – CARACTERÍSTICAS DAS VÁRIAS NORMAS DE REPRESENTAÇÃO GEOMÉTRICA

	Especificação Pública / Gratuita	Informação 3D – objectos / 3D – superfícies / CAD / Visualização	Normalização Internacional / Regional / Consórcio
3D XML	●/●	●/●/●/●	-/-/-
ACIS (SAT/SAB)	●/-	-/●/-/-	-/-/-
DGN	●/-	●/●/●/-	-/-/-
DWF	●/●	●/●/-/●	-/-/-
DWG/DXF	-/-	●/●/●/-	-/-/-
IGES	●/●	-/●/●/-	-/●/-
JT	●/●	●/●/●/●	-/-/-
Parasolid XT	●/●	-/●/-/-	-/-/-
PDF	●/●	●/●/-/●	●/-/-
SLC	●/-	-/●/-/-	-/-/-
STEP	●/-	●/-/-/-	●/-/-
STL	●/-	-/●/-/-	-/-/-
U3D	●/●	●/●/●/●	-/-/●
X3D	●/●	●/●/●/●	●/-/-

Esta tabela apresenta a síntese de algumas das características destes formatos. Mas, para facilitar a análise comparativa, o recorrer a uma forma gráfica torna-se mais adequado. Para se proceder à construção do gráfico é necessário definir qual o tipo de gráfico a utilizar bem como o peso que cada uma das características acarreta.

O tipo de gráfico escolhido para esta análise é o de “Áreas empilhadas”, pois permite ter uma visão incremental sobre as várias características, permitindo analisar o resultado da junção de todas. Segundo o autor, como nenhuma característica se destacada de outra, todas irão ter a mesma contribuição para o gráfico, i.e. todas vão valer a unidade.

Em relação aos diversos parâmetros, visto estes terem várias combinações possíveis é necessário definir a contribuição de cada uma para a apreciação global das características. Começando pela especificação, ambos os valores (Pública e Gratuita) contribuem com o mesmo peso, que se traduz na seguinte equação de resultado normalizado.

$$\text{Especificação} = \frac{(\text{Pública}) + (\text{Gratuita})}{2}$$

No caso da “Informação”, são atribuídos pesos às quatro classes analisadas, ordenadas pela ordem de relevância para a visualização. Começando pelo nível menos relevante, “3D – superfícies” leva peso um, “3D – objectos” peso dois, CAD peso três e a visualização por ser a mais relevante leva peso quatro, como demonstrado na seguinte equação normalizada.

$$\text{Informação} = \frac{4 \cdot (\text{Visualização}) + 3 \cdot (\text{CAD}) + 2 \cdot (\text{3D – objectos}) + (\text{3D – superfícies})}{10}$$

O último parâmetro onde é necessário definir os pesos é a “Normalização”, onde o nível de independência é recompensado. Como aqui não existem valores em simultâneo, consoante a regulamentação existente é dado o valor da contribuição desta característica para o formato. O “Internacional” vai valer a unidade, por ser o mais abrangente, “Regional”, dois terços e o “Consórcio”, um terço. Que se traduz na seguinte equação.

$$\text{Normalização} = (\text{Internacional}) + \frac{2}{3} \cdot (\text{Regional}) + \frac{1}{3} \cdot (\text{Consórcio})$$

Visto todos os parâmetros estarem normalizados, aplica-se as equações definidas na construção do gráfico apresentado na figura seguinte.

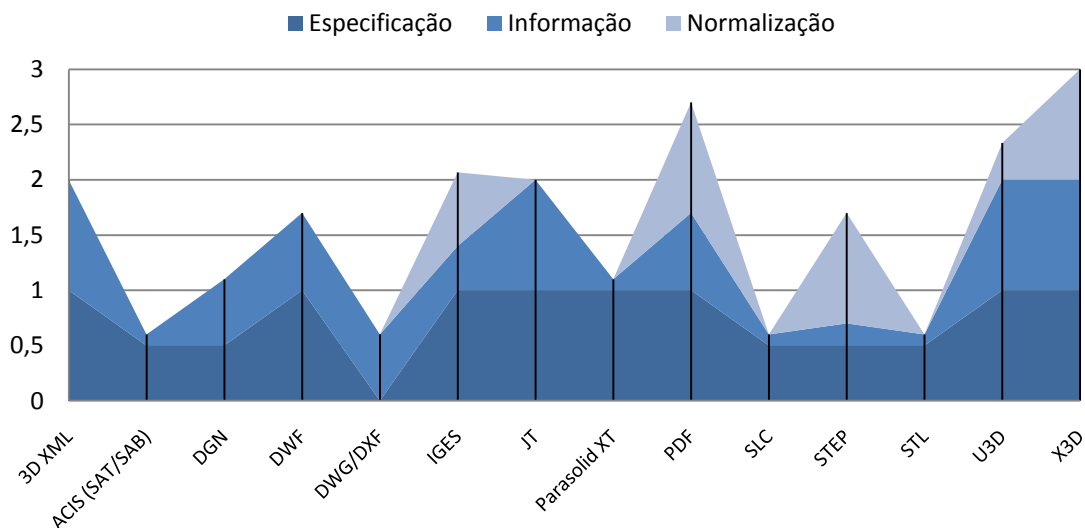


FIGURA 2.3 – GRÁFICO COMPARATIVO DAS NORMAS DE REPRESENTAÇÃO ESTUDADAS.

Avaliando o gráfico, pode-se, através da análise comparativa, verificar como as várias normas respeitam ou não os requisitos definidos anteriormente, concluindo-se que a norma de representação que mais se adequa para ser uma referencia na troca de dados para a visualização, é o X3D. Além de ser a norma que tem melhor resultado, esta cumpre o pleno dos requisitos, obtendo a classificação máxima possível. Apesar de esta ser a que obteve a classificação máxima, não é a única apta para este fim, sendo possível utilizar qualquer uma das outras, podendo por vezes depender da utilização desejada, e.g. o PDF pode ser mais adequado para a criação de manuais técnicos, onde se podem incluir as animações directamente nos documentos.

### 3. VISUALIZAÇÃO SUPORTADA EM MODELOS

A emergência do paradigma da visualização dita que este seja implementado na indústria de uma forma rápida, e sem provocar alterações aos processos e sistemas existentes. Para tal, as soluções para a integração da visualização no ciclo de vida do produto devem ser efectuadas de uma forma transparente, Possibilitando, por exemplo, aos criadores de software a integração da visualização nas suas ferramentas, independentemente das plataformas e linguagens de programação utilizadas, i.e. de uma forma interoperável com os ambientes de trabalho existentes.

Para a realização de uma solução para a visualização, é necessário escolher um método de modelação que permita o desenvolvimento de soluções de software a um nível abstracto sem se recorrer directamente à fase de programação. A metodologia que, hoje em dia, conta com forte investigação associada, propõe a criação de soluções de forma abstracta e independente de tecnologias, e dá pelo nome de Desenvolvimento Orientado pelos Modelos (MDD, do inglês, Model Driven Development).

O MDD é uma importante mudança no paradigma do desenvolvimento de software, pois apresenta uma forma significativamente distinta das utilizadas classicamente para se encarar a complexidade do desenvolvimento de software, tornando-o mais simples e mais formal.

O Desenvolvimento Orientado pelos Modelos é um dos pilares de um outro conceito emergente na indústria, as já atrás referidas Organizações Suportadas em Modelos (MBE) (PDES, Inc., 2007). A metodologia MDD adequa-se ao desenvolvimento de uma solução para a visualização, devido a permitir por exemplo, a interacção com profissionais ligados à indústria na fase do desenvolvimento da solução (e.g. pode-se utilizar na abstracção da solução, conceitos e termos da indústria, quebrando as barreiras técnicas presentes entre os profissionais da indústria e os do desenvolvimento de software), permitindo que estes tenham um papel activo no processo, adaptando a realização às suas necessidades. Outro factor é a facilidade de integração nos sistemas já existentes, através da automatização da criação de código executável em uma qualquer linguagem. (Schmidt, 2006).

---

### 3.1. DESENVOLVIMENTO ORIENTADO PELOS MODELOS

A necessidade da abstracção dos sistemas não é recente. A própria evolução dos métodos, linguagens e tecnologias vem elevando o nível de abstracção de forma a melhorar a qualidade e produtividade do processo de desenvolvimento de software. Iniciativas como a criação do *assembly* para a abstracção do código máquina, o desenvolvimento de linguagens de terceira geração que são compiladas directamente em código de máquina e as linguagens orientadas aos objectos que incluem classes e objectos para abstrair as implementações, são manifestações de uma tendência para a criação de um elevado nível de abstracção, sendo o desenvolvimento orientado pelos modelos, a mais recente inovação para se atingir um nível de abstracção superior.

Fazer a modelação dos sistemas utilizando termos e conceitos familiares aos profissionais que trabalham no domínio do problema, em detrimento de termos apenas familiares aos técnicos conhecedores de detalhes de implementação, permite reduzir a distância existente entre as áreas de negócios e a tecnologia, que é considerado um dos maiores problemas actualmente na inclusão das novas tecnologias no mundo dos negócios (Schmidt, 2006).

No MDD, o artefacto principal é, tal como o nome indica, a noção de modelos. Mas nesta nova visão, o seu papel deixa de ser meramente “contemplativo” (e.g. utilizado em documentação), para passar a ser produtivo (e.g. utilizado na descrição e execução de processos) de forma a ter um papel central no desenvolvimento de software. Continuando-se com o princípio básico do seu homólogo (i.e. o princípio de tudo ser visto como um objecto nas linguagens orientadas pelos objectos), esta forma de modelação rege-se por um princípio de unificação onde “tudo é um modelo”, ou seja, programas, componentes, plataformas, sistemas legados, etc. são todos vistos como modelos, promovendo a simplicidade, generalidade e o poder de integração (Langlois & Exertie, 2005).

Além do princípio da unificação, outro elemento chave no desenvolvimento orientado pelos modelos é o suporte para diferentes níveis de abstracção, desde o alto nível, o dos modelos de negócios focados nos objectivos, papéis e responsabilidade dos intervenientes, até ao nível do detalhe, sendo específico para os casos de uso em concreto e os cenários para a execução do modelo de negócio. Esta propriedade permite trabalhar o problema num nível abstracto com termos e conceitos relacionados com a indústria, e ao mesmo tempo ter uma

---

descrição de como essa abstracção pode ser realizada, de forma a permitir sua execução, automatizando o processo de implementação (Delgado, 2008).

### 3.1.1. MODELOS, META-MODELOS E LINGUAGENS DE MODELAÇÃO

No contexto do desenvolvimento orientado pelos modelos, o modelo é um conjunto de descrições que definem um sistema em estudo, i.e. define os elementos, suas relações e comportamento, que realizam o sistema. Os modelos têm de ser descritos utilizando uma linguagem de modelação bem definida, tanto em termos de sintaxe como de semântica, de forma a poder ser interpretada automaticamente por computador.

A linguagem de modelação utilizada na descrição dos modelos é também ela própria descrita num modelo. Este modelo, referido como meta-modelo, consiste num modelo que define as construções e as relações que podem ser utilizadas numa dada linguagem de modelação. Assim, uma linguagem de modelação resulta da aplicação das descrições definidas no meta-modelo. Por exemplo, na linguagem EXPRESS (que é utilizada para definir modelos ISO10303) são definidos construtores (e.g. “Entity”, “Type”, “Where”, etc) que são utilizados para se criarem modelos EXPRESS válidos. Como referido, o meta-modelo sendo também um modelo, necessita de uma linguagem que o descreva. Esta linguagem é chamada de meta-linguagem e pode ser considerada uma linguagem especializada para descrever linguagens de modelação

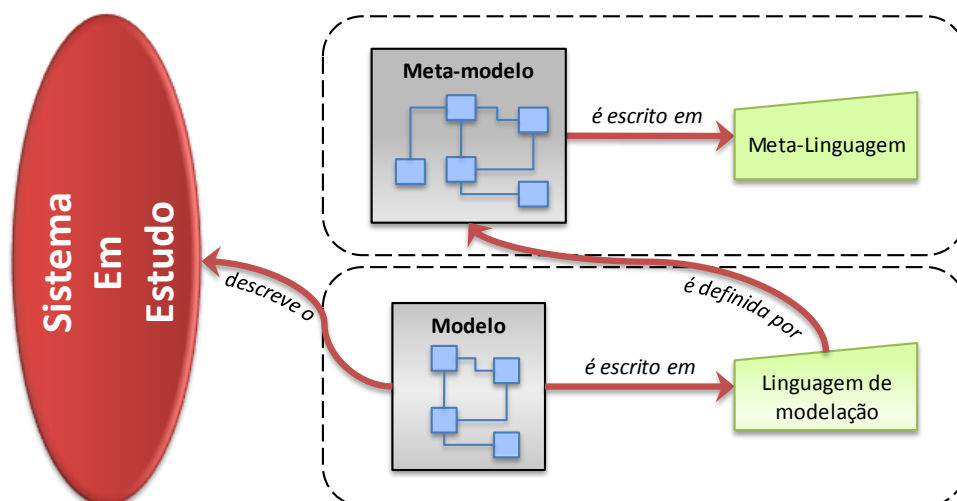


FIGURA 3.1 – RELAÇÃO ENTRE MODELOS, META-MODELOS E LINGUAGENS DE MODELAÇÃO.

Para se descrever um sistema, em teoria, seriam necessárias infinitas camadas de modelos, linguagens de modelação, meta-modelos e meta-linguagens, pois para se definir um modelo,

é sempre necessário um meta-modelo associado. Para superar esta barreira, existe o conceito de meta-modelos reflexivos, que são expressos na mesma linguagem que eles próprios definem, i.e. tornando-se o seu próprio meta-modelo. Exemplos de meta-modelos reflexivos são o OMG Meta Object Facility (MOF) e o Ecore.

Todavia, de forma a realizar operações entre modelos, como por exemplo, comparar, transformar, combinar, associar, etc. é necessário estabelecer um número finito de camadas que permita regulamentar os processos. Para tal adoptou-se a aproximação efectuada pela OMG, que especificou uma arquitectura que limita o número de camadas a quatro. Esta arquitectura consiste numa hierarquia de modelos, onde cada modelo de uma camada é definido como uma instanciação do modelo da camada anterior, excepto o da camada de topo que é do tipo reflexivo, i.e. que se define a ele próprio.

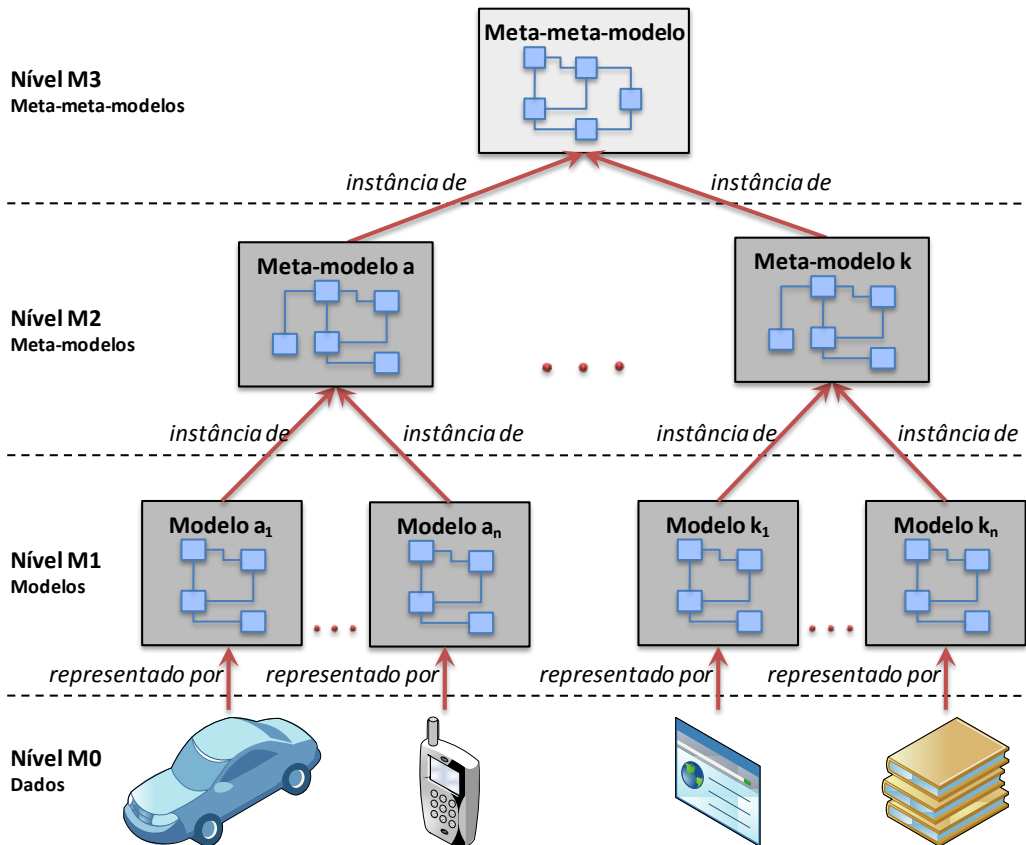


FIGURA 3.2 – ARQUITECTURA DE MODELAÇÃO SEGUNDO A OMG.

Utilizando a figura como guia, a camada inferior, M0, representa os dados a ser manipulados pelo sistema, a camada M1, representa os modelos que definem como a informação da camada dos dados é especificada. Entrando nas camadas dos meta-modelos, a camada M2, representa os meta-modelos que definem as linguagens de modelação utilizadas na



definição dos modelos, sendo estes definidos pela camada superior, chamada de meta-meta-modelos. Esta última contém apenas um meta-modelo do tipo reflexivo que se comporta como o nível superior de modelação, e garante a possibilidade de encontrar relações entre todos os modelos existentes na representação do sistema.

Outro aspecto na modelação orientada pelos modelos é a possibilidade de se representar o mesmo sistema com modelos definidos em vários níveis de abstracção. Neste contexto uma abstracção representa uma operação que simplifica um modelo, suprimindo detalhes irrelevantes para o nível de abstracção desejado (e.g. apresentar processos de negócios). O contrário da abstracção é o refinamento, que consiste em incluir detalhes mais concretos (e.g. detalhes técnicos de execução). A figura seguinte ilustra estas noções.

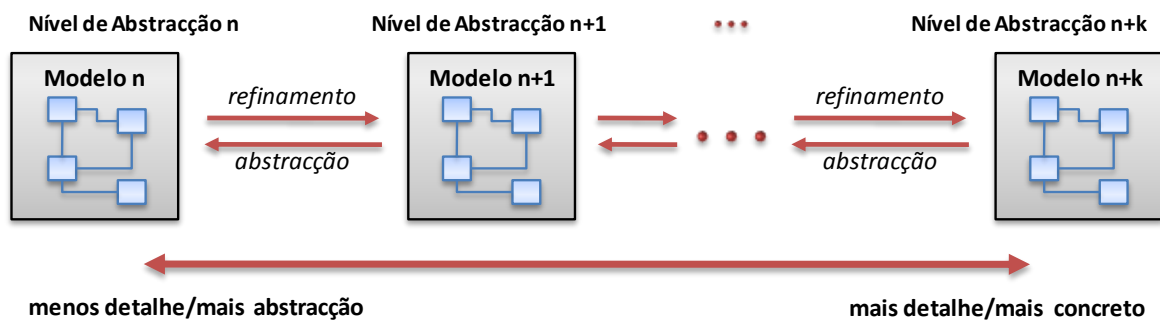


FIGURA 3.3 – OPERAÇÕES DE REFINAMENTO E ABSTRACÇÃO EM MODELOS.

Este processo de abstracção e refinamento aumenta a possibilidade de existirem mais profissionais envolvidos no processo de modelação de um sistema. Por exemplo, num alto nível de abstracção, os profissionais que estão mais próximos do sistema, podem interagir na definição dos modelos modelando os processos necessários, ou seja, podem contribuir na definição dos requisitos ao nível do negócio. Para esses requisitos serem implementados, tem de se recorrer a profissionais de um nível mais técnico para se introduzir detalhes de execução dos processos, para se poder chegar a um nível em que o sistema possa ser implementado e executado.

Com a existência de várias linguagens e formas diferentes de modelação, cria-se a necessidade de se criar um processo de transformação entre modelos, onde aplicando um conjunto de regras de transformação, definidas numa linguagem especial, chamada de linguagem de transformação (e.g. ATL – ATLAS Transformation Language), se transformam

modelos em modelos. Este tema é abordado com mais detalhe no subcapítulo seguinte (Delgado, 2008).

### 3.1.2. MORFISMOS DE MODELOS

Morfismo é um termo utilizado na matemática, mais concretamente no campo teórico que trabalha, de forma abstracta, com estruturas matemáticas e relações entre elas. Neste contexto, o morfismo é uma abstracção dos mapeamentos entre duas estruturas, normalmente conjuntos de objectos, podendo ser visto como uma função que associa qualquer tipo de objectos, onde o conjunto de origem é chamado de domínio e o de destino chamado de co-domínio.

Este termo foi recentemente importado para o mundo da modelação onde é utilizado na definição de um novo conceito, o morfismo de modelos, conhecido como MoMo, sigla que deriva da expressão em inglês “Model Morphism”. Este conceito foi introduzido como o resultado de investigação do projecto internacional INTEROP-NoE<sup>4</sup>, e tem como objectivo a representação de todo o tipo de operações entre modelos, ou seja, dados dois modelos, fonte e alvo, um MoMo entre eles representa uma operação que aplicada ao primeiro resulta no segundo, como ilustrado pela figura seguinte.



FIGURA 3.4 – MORFISMO DE MODELOS.

De modo a facilitar a utilização e a definição de MoMos, o consórcio INTEROP-NoE desenvolveu um sistema de recomendação de MoMos, que actua como um repositório de conhecimento para os MoMos. Com este sistema é possível verificar as soluções existentes para os problemas encontrados anteriormente, sendo disponibilizado através de um portal Web e para, facilitar a navegação por parte dos utilizadores, foi definida uma ontologia, que classifica as soluções MoMo existentes.

---

<sup>4</sup> INTEROP-NoE – <http://www.interop-noe.org/>

---

Alguns dos conceitos definidos pela ontologia dos MoMos são os seguintes:

- ModelOperation: qualquer tipo de manipulação que possa ser efectuada a um ou mais modelos, como a criação de modelos ou o processamento dos mesmos.
- ModelCreation: passos empreendidos durante o processo da criação de modelos.
- ModelProcessing: operações que se podem fazer aos modelos depois de eles terem sido criados, como a descoberta de morfismos e a transformação de modelos.
- ModelMorphismDiscovery: operação que consiste em descobrir quais os mapeamentos existentes entre dois ou mais modelos.
- ModelTransformation: operação que, dado um ou mais modelos, retorna o modelo processado após ter sido efectuada uma transformação.
- EnablingTechnology: tecnologia onde se realiza as operações aos modelos, seja a metodologia utilizada, a ferramenta de software ou outra.
- Purpose: razão para a qual as transformações são efectuadas, seja para juntar modelos, transformá-los noutros, etc.
- Approach: métodos ou metodologias utilizadas para a realização das operações, como os grafos, linguagens de programação, etc.

Dentro dos conceitos definidos na ontologia dos MoMos, os mais relevantes para a harmonização de modelos, um dos objectivos deste trabalho, são a descoberta de morfismos e a transformação de modelos, que portanto irão ser descritos de seguida em maior pormenor.

#### 3.1.2.1. DESCOBERTA DE MORFISMOS

A descoberta de morfismos é uma operação, que não provoca alterações aos modelos envolvidos, e consiste em, dados dois modelos descobrir as correspondências existentes entre eles, i.e. esta operação retorna as relações existentes entre eles sem os alterar.

As correspondências são portanto as relações entre elementos de dois modelos, e podem ser vistas como as regras que descrevem a forma como um ou mais elementos de um modelo fonte podem ser relacionados com um modelo alvo, existindo três tipos de correspondências possíveis entre modelos:

---

- $1 \rightarrow 1$ , um elemento do modelo fonte corresponde exactamente a um elemento do modelo alvo;
- $1 \rightarrow N$ , um elemento do modelo fonte tem correspondência a vários elementos do modelo alvo;
- $N \rightarrow M$ , um grupo de elementos do modelo fonte tem correspondência com outro grupo de elementos do modelo alvo.

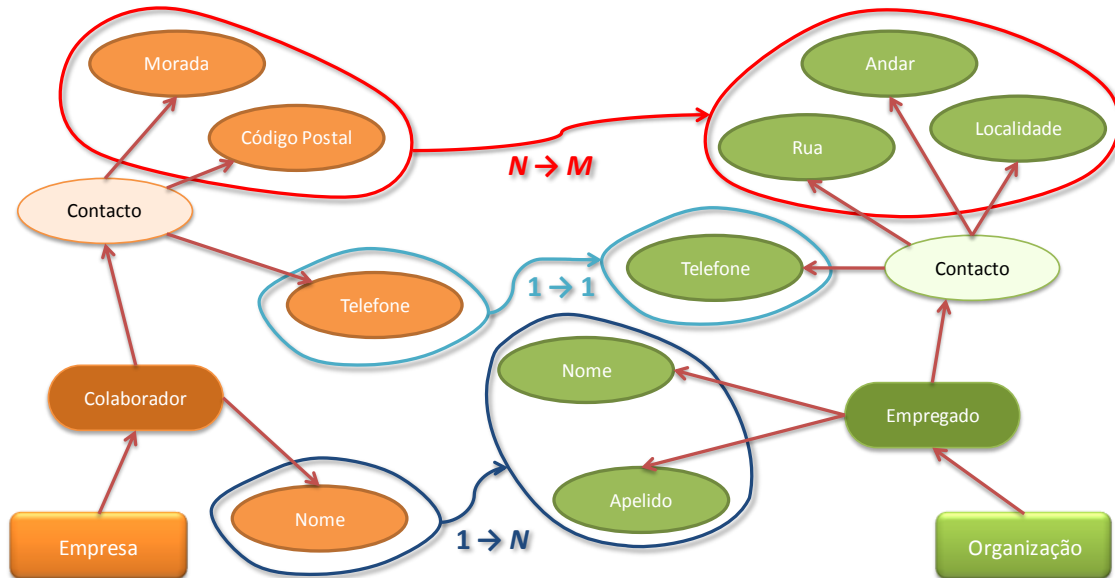


FIGURA 3.5 – DIFERENTES TIPOS DE CORRESPONDÊNCIAS ENTRE DOIS MODELOS.

O resultado da descoberta de morfismos é chamado de mapeamento, e corresponde à definição de todas as relações existentes entre dois modelos. Este resultado pode ser classificado como unidireccional ou bidireccional, dependendo se o mapeamento é válido apenas em uma direcção ou em ambas, respectivamente. Por sua vez, numa outra perspectiva, os mapeamentos podem ser classificados em dois tipos: mapeamento ao nível dos tipos e ao nível das instâncias.

Um mapeamento ao nível das instâncias define a correspondência entre os elementos dos modelos, sendo normalmente definido, como é que os elementos devem ser transformados uns nos outros. Por exemplo, no caso apresentado na Figura 3.5, ao nível das instâncias seria definido como a Morada e Código Postal se mapeavam na Rua, Andar e Localidade, sendo que os tipos que estes utilizam serão mapeados nos mapeamentos de tipos.

No mapeamento por tipos, a correspondência é efectuada ao nível do tipo dos elementos e não dos elementos em si. Este mapeamento é independente dos modelos definidos, sendo

efectuado nos seus meta-modelos a um nível mais abstracto. Todas as instâncias de um certo tipo no modelo fonte correspondem às instâncias de um outro tipo no modelo de destino, processando-se ou não a informação nele descrita, como por exemplo o mapeamento entre duas linguagens utilizadas na modelação, o XML e o UML, onde os elementos do XML podem ser correspondidos às classes do UML.

As relações ao nível dos meta-modelos ocorrem de forma análoga às relações entre modelos, visto que os próprios meta-modelos são modelos (Delgado, 2008).

#### 3.1.2.2. TRANSFORMAÇÃO DE MODELOS

Ao contrário da operação de descoberta de morfismos, a transformação de modelos é uma operação que modifica o modelo fonte noutro, sendo que ambas as operações estão intrinsecamente relacionadas, i.e. de forma a se realizar uma transformação, um mapeamento tem de estar definido, seja ele implícito ou explícito. De facto, são as regras de correspondência definidas num mapeamento que descrevem como um modelo pode ser transformado noutro, sendo portanto a descoberta de morfismos um pré-requisito para a transformação. Pode então a transformação de modelos ser vista como uma operação que provoca uma alteração aos modelos, e consiste em processar um ou mais modelos com base na especificação de mapeamento, retornando o modelo desejado.

Como referido na descoberta dos morfismos, o mapeamento pode ser descrito em dois níveis, o de instâncias e o de tipos. A transformação herda esta classificação, sendo classificadas as transformações em: transformações de instâncias (recorrendo-se ao mapeamento de instâncias) e transformações orientadas pelos meta-modelos (utilizando o mapeamento de tipos).

No caso da transformação de instâncias, a transformação é aplicada directamente aos dados que os modelos onde o mapeamento foi definido representam; no caso das transformações orientadas pelos meta-modelos, a transformação não acontece directamente nos dados que o sistema representa, mas sim ao nível dos modelos representados pelos meta-modelos definidos. Por exemplo, no caso da arquitectura de modelação proposta pela OMG apresentada anteriormente (Figura 3.2), o mapeamento é estabelecido ao nível dos meta-modelos, nível M2, e a transformação propriamente dita ocorre no nível dos modelos, nível M1, como pode ser observado em forma de diagrama na figura seguinte.

---

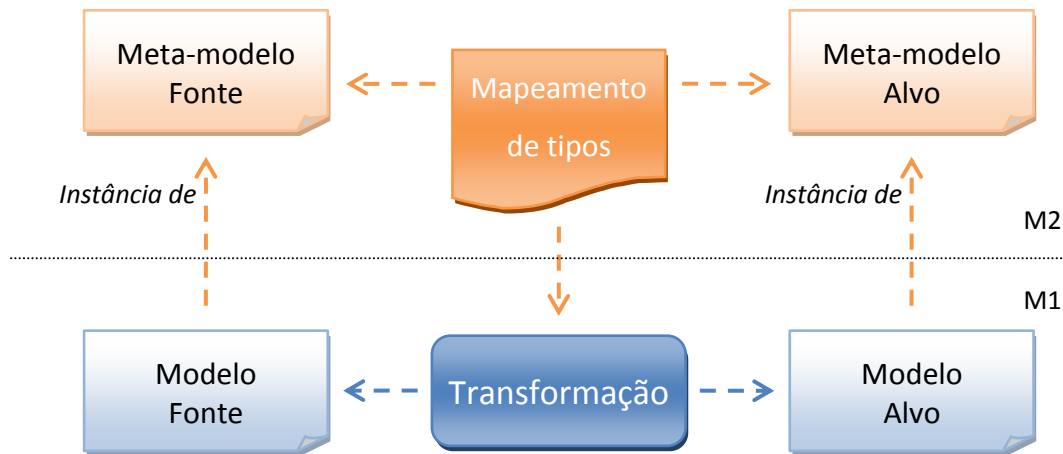


FIGURA 3.6 – TRANSFORMAÇÃO ORIENTADA PELOS META-MODELOS.

A escolha entre os dois métodos de transformação depende do nível de flexibilidade desejado. Por exemplo, o caso da transformação orientada pelos meta-modelos é mais determinista do que a transformação de instâncias, visto ser baseada no bem definido mapeamento de tipos entre os meta-modelos envolvidos. No caso de existir necessidade de transformar informação adicional específica às instâncias dos meta-modelos é mais adequada a utilização da transformação de instâncias.

Adicionalmente existe outro aspecto que influencia qual a escolha a efectuar para o tipo de transformação a ser efectuada, que é se o modelo alvo e de origem derivam ambos do mesmo meta-modelo. Para tal, existe mais uma forma de se classificar as transformações, as endógenas, para modelos derivados do mesmo meta-modelo ou exógenas, quando são derivados de meta-modelos diferentes.

As transformações orientadas pelos meta-modelos são mais adequadas às transformações endógenas, uma vez que tanto o modelo original e de destino estão descritos pelas mesmas restrições definidas num meta-modelo comum, facilitando a validação das transformações. No caso das transformações exógenas, os mapeamentos orientados as instâncias são mais adequados uma vez que as regras e as restrições que definem os modelos são diferentes para cada modelo envolvido.

Todavia, em alguns casos uma combinação dos dois tipos de mapeamento oferece uma melhor solução, uma vez que permite utilizar a flexibilidade do mapeamento de instâncias, e ao mesmo tempo utilizar o mapeamento de tipos para preservar as restrições de tipos definidas nos meta-modelos (Delgado, 2008).

### 3.2. VISUALIZAÇÃO ORIENTADA PELOS MODELOS

Hoje em dia existem inúmeras ferramentas para a aplicação de métodos de visualização. Apesar de estas serem baseadas em interfaces bastante evoluídas, sofrem da falta de flexibilidade e adaptabilidade para serem aplicadas de uma forma interoperável no mundo industrial. É um facto bem conhecido que as interfaces visuais se tornam mais efectivas quando personalizadas e adaptadas para o domínio de utilização. É também sabido que as soluções para aplicação da visualização, quando integradas nos ambientes de trabalho dos profissionais, permitem uma maior utilização e aceitação por parte dos mesmos (MacKrell, *Extracting Value from Your Product and Process Data*, 2006).

A forma corrente de se desenvolver aplicações de visualização passa pela fase de programação, implementando-se directamente a forma como a informação é apresentada. Além deste ser um método que peca pela falta de personalização, ainda conta com uma difícil integração com as ferramentas já existentes. Conduzir a operação da criação de soluções para a visualização a um nível mais abstracto, permite que os profissionais que fazem uso da informação possam ter um papel activo no desenvolvimento destas soluções, podendo obter vistas totalmente orientadas às suas necessidades, utilizando o seu saber na criação das mesmas. Para esta abstracção o método atrás descrito, o desenvolvimento orientado pelos modelos, enquadra-se nas soluções para a transposição desta barreira (Bull & Favre, *Visualization in the Context of Model Driven Engineering*, 2005).

A aplicação do desenvolvimento orientado pelos modelos à visualização passa pela construção de modelos que representem as estruturas utilizadas pelos métodos de visualização, permitindo a aplicação de morfismos de modelos, entre os modelos do sistema e os da visualização. Este conceito, da aplicação do desenvolvimento orientado pelos modelos ao paradigma da visualização, resulta no emergente paradigma de desenvolvimento, que dá pelo nome de visualização orientada pelos modelos. Com a sua emergência, e pela sua aceitação, existem arquitecturas que definem a aplicação deste paradigma, sendo a arquitectura de referência apresentada na figura seguinte (Bull, Storey, Favre, & Litoiu, 2006).

---

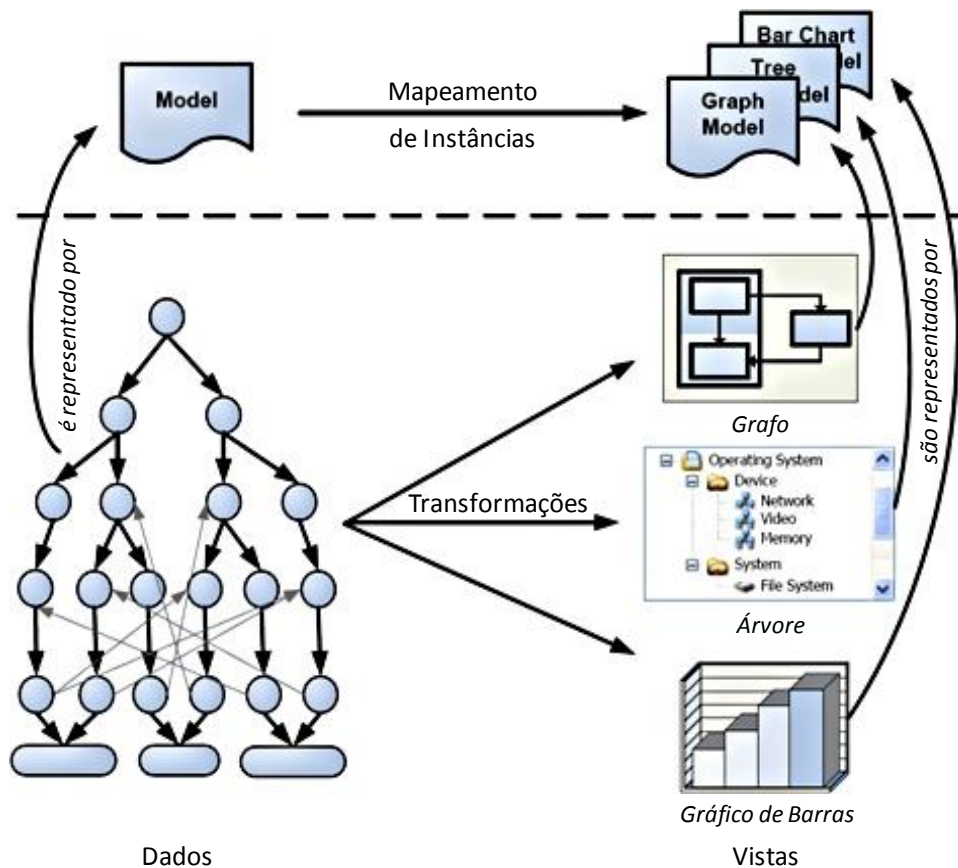


FIGURA 3.7 – ARQUITECTURA DE REFERÊNCIA PARA A VISUALIZAÇÃO ORIENTADA PELOS MODELOS ADAPTADA DE (BULL, STOREY, FAVRE, & LITOIU, 2006).

Estas várias formas de visualização contribuem de uma forma mais eficaz caso se adaptem entre si. Na figura anterior, os dados são transformados em vistas diferentes, mas estas vistas não contêm nenhuma associação entre elas, sendo que ao se integrar as várias vistas se pode ter uma visualização mais efectiva, permitindo, por exemplo, para se associar um mapa geográfico e uma lista de localizações, as escolhas efectuadas na lista devem ser reflectidas no mapa, i.e. aquando a selecção do item da lista, o mapa geográfico deve de ser alterado para a localização pretendida. Logo, as várias vistas teriam de ter mapeamentos definidos tanto com os dados, como entre si, de forma a se adaptarem umas com as outras (Bull, Integrating Dynamic Views Using Model Driven Development, 2006).

### 3.2.1. VISTAS ORIENTADAS PELOS MODELOS

A aplicação do paradigma da visualização passa pela adopção de diversas vistas sobre os dados do sistema, de forma a facilitar a sua interpretação. As vistas são parte integrante do paradigma da visualização, são estas que o aplicam, pois descrevem os artefactos visuais utilizadas na apresentação da informação. As vistas são portanto o meio para a aplicação da



visualização. Com a sua aplicação aos dados consegue-se simplificar o seu conteúdo mostrando-se apenas o necessário para desenrolar das funções dos profissionais, melhorando o seu desempenho.

As vistas podem existir de diversas formas, desde as mais simples (e.g. grafos, tabelas, etc.) até às mais complexas (e.g. mapas, mostradores tridimensionais, etc.), tendo em comum tanto a sua contribuição para uma maior objectividade na apresentação da informação, como a forma com que se ligam ao desenvolvimento orientado por modelos.

O desenvolvimento orientado por modelos, nos moldes atrás referidos (utilizando a arquitectura proposta pela OMG como referência), prevê a existência de quatro camadas para descrever sistemas, e as vistas não são excepção. O importante a referir é, como se efectua a ligação entre as vistas e os modelos de dados, de modo a permitirem toda a automatização do processo de transformação dos dados de um sistema numa vista adequada.

As vistas são definidas por dois aspectos, os dados que representa, e a sua aplicação (i.e. a implementação de como esses dados são representados em tempo de execução). Ao nível da modelação dos problemas, e para permitir os morfismos entre dados e vistas, interessa apenas os modelos de dados utilizados pelas vistas, e não os detalhes de implementação.

Com esta abordagem, a utilização das vistas, na fase de desenvolvimento das aplicações é facilitada, porque por vezes, as vistas mais complexas são de difícil utilização, o que leva a programas mais dispendiosos (devido ao seu maior tempo de desenvolvimento). Com uma boa modelação das vistas, essa complexidade é abstraída, o que possibilita que para a utilização desse tipo de visualizações (as mais complexas), não seja preciso ser-se conhecedor do funcionamento das mesmas, mas apenas do seu modelo, definindo-se os morfismos entre os modelos que descrevem os dados a visualizar e os definidos pelas vistas, ficando a parte técnica das vistas num nível diferente dos modelos (parte onde é efectuado o desenvolvimento) que não é relevante para a utilização das mesmas.

A formalização das vistas passa portanto, pela descrição dos seus modelos de dados, numa linguagem de modelação escolhida. Na figura seguinte é apresentado um exemplo da descrição de duas vistas nos moldes definidos pela arquitectura da OMG. Este exemplo foi retirado de (Faure & Allilaire, 2007), e tem tanto a descrição dos modelos de dados de uma

---

árvore e uma lista numa linguagem de modelação (Ecore, que tem a particularidade de ser comum ao nível M2 e M3, meta-modelos e meta-meta-modelos, respectivamente, visto esta ser do tipo reflexivo e não necessitar de ter um meta-modelo a descreve-la), como a definição dos morfismos entre eles. Neste exemplo em concreto, estes modelos de dados foram aplicados a dois objectos visuais do java, *jtree* (Sun Microsystems, Inc., 2008) e *jlist* (Sun Microsystems, Inc., 2008), e são apresentados na figura seguinte.

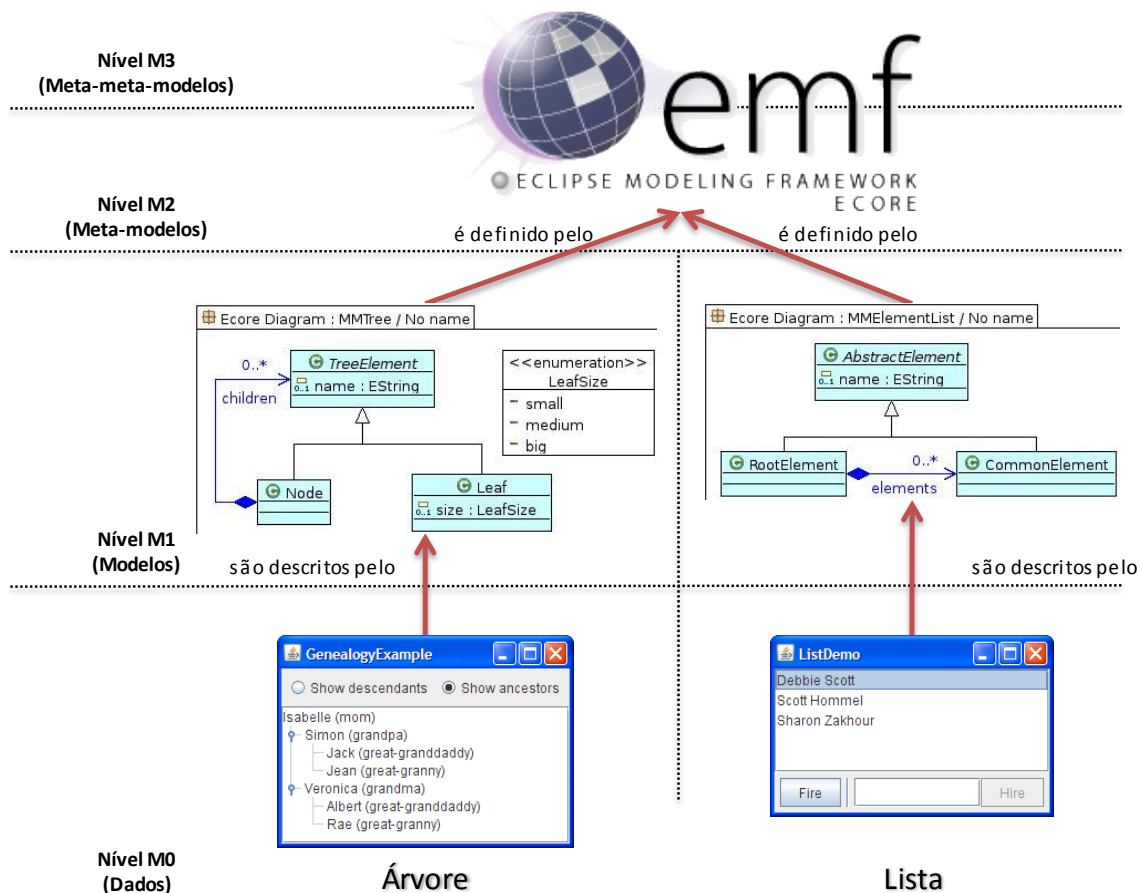


FIGURA 3.8 – APLICAÇÃO DA VISUALIZAÇÃO ORIENTADA POR MODELOS A UMA ÁRVORE E UMA LISTA.

Utilizando as vistas consegue-se aplicar o paradigma da visualização orientada por modelos, conseguindo-se trazer ao mundo da interactividade com os utilizadores as potencialidades do desenvolvimento orientado por modelos, podendo-se fazer mais rapidamente e eficientemente interfaces de uso mais avançadas, promovendo-se uma melhor facilidade de utilização das ferramentas que contam com as vistas para aumentar o seu valor (Bull, Integrating Dynamic Views Using Model Driven Development, 2006).

#### 4. SOLUÇÃO INTEROPERÁVEL PARA A VISUALIZAÇÃO DE DADOS DE PRODUTO

Uma solução neste contexto, dos sistemas das tecnologias de informação e comunicação, é a combinação de várias funcionalidades para solucionar um problema de negócio. Portanto a criação de uma solução adequada e robusta, passa pela percepção de quais os problemas existentes que se pretendem colmatar, para se proceder posteriormente ao desenvolvimento de uma solução que permita resolver os problemas identificados.

Anteriormente no capítulo 2 (Visualização de dados de produto em ambientes industriais) foram identificados os grandes desafios desta dissertação, desafios estes que vão ter um papel preponderante no desenvolvimento da solução para a unificação da visualização de dados de produto, visto estes conduzirem os processos de desenvolvimento que definem qual a resposta aos desafios entrepostos.

O **primeiro desafio** tem a ver com o facto de grande parte dos produtos colocados no mercado serem desenvolvidos recorrendo aos mais diversificados sistemas e competências individuais, e estes serem guardados e trocados nos mais variados formatos e versões. O que leva a que seja imperativo existir um ambiente interoperável onde o formato a utilizar não se torne impeditivo no desenrolar das funções dos intervenientes. Logo, a solução visionada necessita ter **métodos para a informação ser utilizada de forma independente dos formatos de troca**, promovendo a interoperabilidade no ambiente de trabalho.

Passando para o **segundo desafio**, este passa pela existência de inúmeros intervenientes de diferentes domínios, e nem todos necessitam do mesmo tipo de informação para efectuar o seu trabalho. Adaptando a informação ao domínio de aplicação de uma forma simples e de fácil interpretação é uma mais-valia. Portanto, a solução tem de ser capaz de **contextualizar a informação** bem como apresentar a mesma de formas visuais intuitivas para o desenrolar das funções próprias de cada interveniente.

O **terceiro, e último desafio**, deriva do facto de a alteração das ferramentas de trabalho utilizadas pelos profissionais provocarem um revés na eficiência do seu trabalho, aumentando os custos associados, quer nas licenças, quer na formação. Levando a que a solução óptima passe pela utilização apenas de uma ferramenta para o acesso à informação relevante para o desempenho das suas funções. Portanto, é relevante que a **solução possa ser integrada nas ferramentas de trabalho já existentes**.

---

Dos três grandes desafios identificados, os dois primeiros vão levar à decisão de quais os fundamentos funcionais da solução, sendo que o terceiro, não menos importante, tem a sua contribuição na definição da metodologia utilizada na modelação e desenvolvimento desta solução, já estudado anteriormente, o Desenvolvimento Orientado pelos Modelos.

#### 4.1. FUNDAMENTOS DE SUPORTE À SOLUÇÃO

Analisando os desafios atrás lançados, conseguem-se identificar três fundamentos principais que, ao serem incluídos na solução numa forma unificada e utilizados em conjunto, permitem colmatar as necessidades encontradas, bem como abrir o espectro de aplicabilidade e expansibilidade da solução. Estes são a harmonização de formatos, a aplicação de artefactos visuais, e a contextualização da informação.

A harmonização de formatos é relevante, num ambiente heterogéneo, onde não existe garantia que a informação trocada possa ser utilizada por outros sistemas. Portanto possibilitar que os vários sistemas importem e exportem qualquer tipo de ficheiros, promove a interoperabilidade. Integrar a solução para a visualização nos sistemas, permite que esta seja utilizada para transformar qualquer formato de dados, em formatos interpretáveis pelo sistema, permitindo assim a expansibilidade do mesmo, quebrando o limite e barreiras existentes nos formatos suportados. Com esta integração, os vários intervenientes no ciclo de vida do produto não precisam ter em conta quais os sistemas que os outros agentes utilizam, porque podem ter a garantia que estes, recorrendo a esta solução, conseguem utilizar a informação trocada, independentemente do formato utilizado.

A aplicação de artefactos visuais verifica-se no caso de existir informação que se torna de difícil percepção quando não apresentada nos meios visuais adequados. Por exemplo, informação geográfica é de mais fácil percepção quando apresentada num mapa, informação tridimensional se apresentada sem ser num visualizador tridimensional é de difícil interpretação, etc. Para se simplificar a interpretação e compreensão da informação, é necessário que esta seja apresentada de uma forma visual adaptada ao tipo de informação que se está a utilizar, para que melhore a rapidez e facilidade de apreensão da mesma pelos intervenientes, melhorando a eficácia e eficiência no desempenho das suas funções.

A contextualização da informação, permite fornecer aos intervenientes, quer através de um artefacto visual, quer trocando um ficheiro de dados, estritamente a informação que este

---

necessita para desempenhar as suas funções. Esta permite aplicar algumas funcionalidades que podem ser relevantes numa troca comercial, e podem ir desde a possibilidade de se proteger informação, como retirar alguma informação sensível das trocas, até ao tratamento da informação para apresentar visualmente a informação útil para o desempenho das funções dos utilizadores. Por exemplo, um canalizador ao trabalhar na planta de um prédio quer a informação contextualizada à sua função, este não estará interessado em ver a parte eléctrica, os acabamentos, etc., mas sim a estrutura básica do prédio e a canalização.

#### 4.2. CONCEITO DA SOLUÇÃO

O conceito desta solução passa pela análise dos fundamentos atrás definidos, e pela percepção como os mesmos podem ser utilizados como um guia para a concepção da mesma. Numa primeira fase, vai-se definir conceptualmente a solução de modo a começar a abordar o problema de uma forma abstracta. Portanto, para se avançar para o conceito desta solução, é necessário identificar quais os pontos de ligação a entidades externas que os vários fundamentos vão introduzir na solução.

Começando pela harmonização de formatos, e para permitir que se efectue conversões entre formatos, é necessário ter como entradas e saídas os ficheiros descritos nos formatos a utilizar. No caso da aplicação de artefactos visuais, as saídas que a solução vai promover, são os dados para os diversos artefactos visuais. Por último, e para existir contextualização de informação, é necessário que a solução conheça o contexto a utilizar, por isso, esta deve receber uma especificação que defina esse contexto.

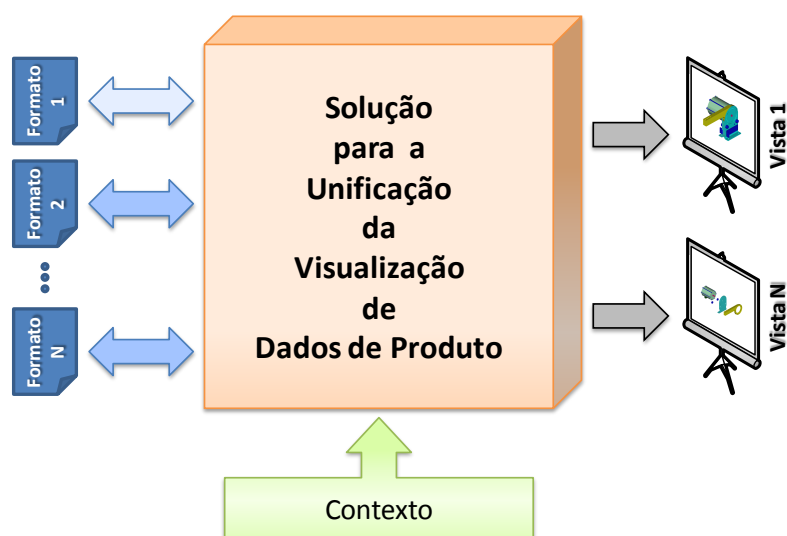


FIGURA 4.1 – SISTEMA PARA UNIFICAÇÃO DA VISUALIZAÇÃO DE DADOS DE PRODUTO.

Depois de definidos quais os pontos de ligação a entidades externas que os fundamentos incluem na solução, é necessário perceber como os mesmos se interligam entre si já dentro da própria solução. Um dos problemas que existe quando se tem em simultâneo várias componentes (formatos, contextualização e artefactos visuais) e estas têm dentro delas próprias várias formas distintas de existir (i.e. existem vários formatos e artefactos visuais), é na forma como se efectua a ligação entre elas.

Na figura seguinte é apresentado o cenário típico da ligação de N para M elementos. De notar que o número de ligações iria aumentar exponencialmente com a inclusão de novos formatos, ou artefactos visuais, e a contextualização de informação iria necessitar ter uma aplicação específica para cada base de formato existente, que é deixada de parte neste exemplo por uma questão de simplificação do diagrama.

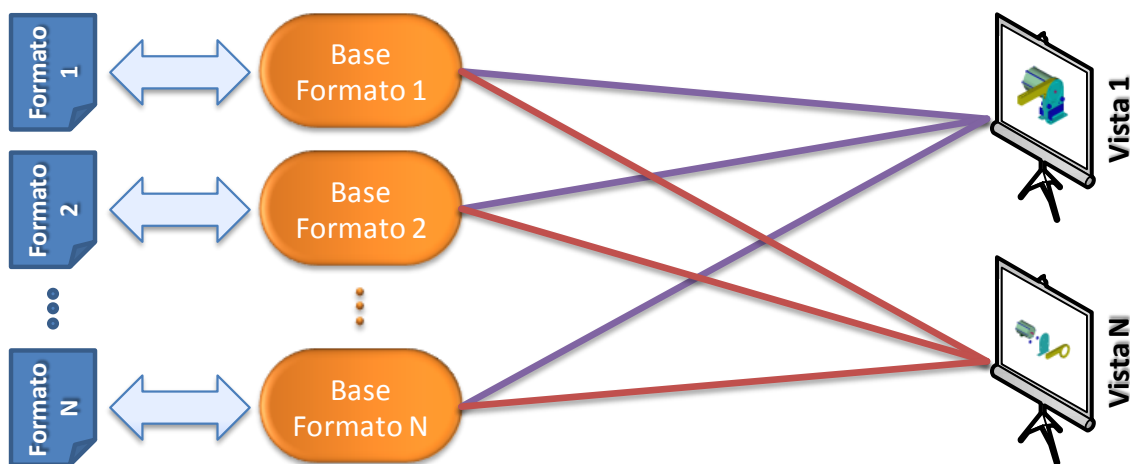


FIGURA 4.2 – DEMONSTRAÇÃO DO NÚMERO DE LIGAÇÕES ENTRE FORMATOS E ARTEFACTOS VISUAIS.

Dado o problema das ligações crescerem de forma exponencial, é necessário encontrar uma forma de limitar o número de ligações, bem como facilitar a futura inclusão de formatos, a inclusão de artefactos visuais e a aplicação de novas regras de contextualização. Para tal vai-se basear a solução no modelo unificado de interoperabilidade (ISO 14258, 1999).

A abordagem unificada de interoperabilidade assume que existe um modelo que fornece uma estrutura comum aos vários modelos constituintes, criando um meio para estabelecer equivalência semântica entre eles. Usando esta estrutura comum, qualquer modelo pode ser traduzido em qualquer outro, apesar de não se poder garantir que não ocorra perda de semântica. Portanto, apelidando este modelo comum de base comum, esta é onde se define

uma parte comum aos artefactos visuais e formatos existentes, e onde as regras de contextualização de informação são aplicadas de uma forma única.

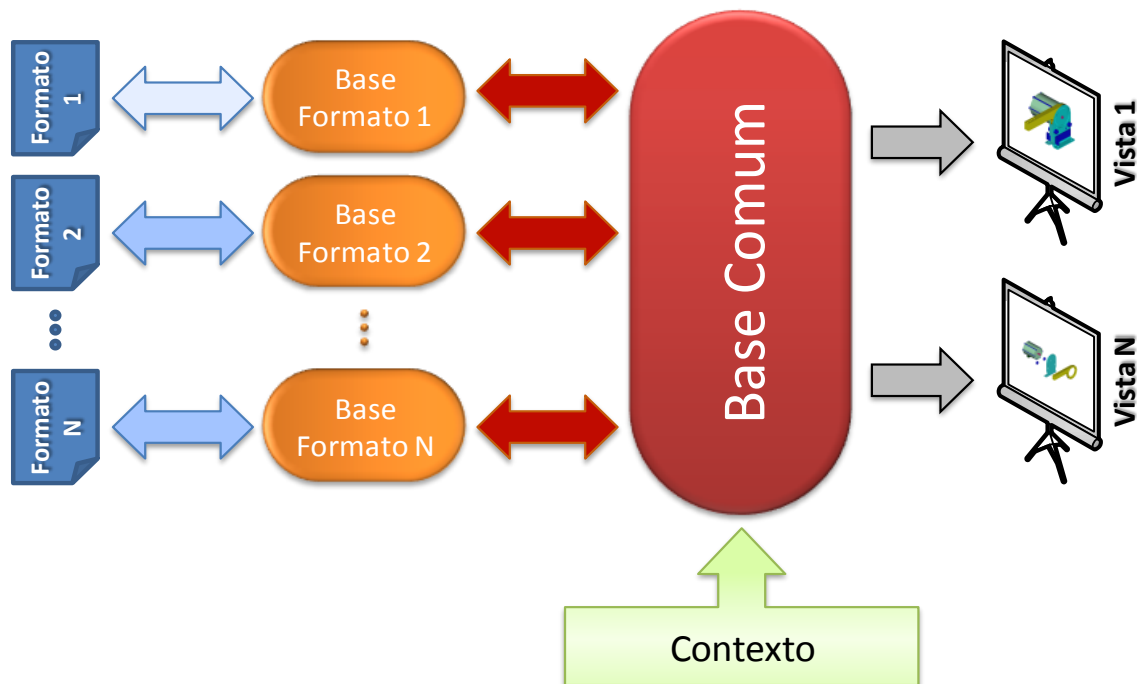


FIGURA 4.3 – ESPECIFICAÇÃO DA SOLUÇÃO.

Uma vantagem da utilização desta passa pela facilidade de implementação da solução, ou seja, evita-se que seja necessário definir como é que se criam as vistas e se contextualiza a informação para todos os formatos existentes, passando estas a ser definidas através de uma base comum, sendo apenas necessário definir uma vez como se criam as vistas e como se contextualiza a informação.

Outra vantagem associada à utilização de uma base comum é o facto de toda a solução para a importação/exportação dos ficheiros físicos de cada formato poder ser desenvolvida por entidades distintas e integradas na solução. Sendo o grande obstáculo desta abordagem a necessidade de fazer uma ligação entre os vários formatos e a base comum atrás descrita, cujas ligações são apresentadas de uma forma geral na figura seguinte.

Esta é portanto a especificação abstracta da solução, onde são apresentadas as várias relações entre as diversas componentes conceptuais da solução, permitindo que se vá ao encontro dos fundamentos identificados. Mas, para esta solução passar de um conceito para uma solução utilizável (i.e. implementável e executável), é necessário especificar como é que as diversas componentes e suas relações podem ser descritas formalmente.

### 4.3. ARQUITECTURA DA SOLUÇÃO

Uma arquitectura é a definição formal de um sistema, organizada de forma que demonstre as propriedades estruturais do mesmo. Definindo quais os seus componentes ou blocos e as relações entre eles existentes, que possam ser desenvolvidos ou reaproveitados de forma a implementar o sistema.

Portanto, a aplicação do desenvolvimento orientado pelos modelos a esta solução permite a definição da arquitectura formal, para que esta possa ser modelada e implementada para uma futura utilização (no caso concreto deste trabalho, uma prova de conceito), passando esta pela definição de onde se aplicam os modelos e quais os morfismos necessários.

Então, para simplificar o processo de desenvolvimento da arquitectura orientada pelos modelos, vai-se em primeiro lugar, definir onde os modelos se enquadram e qual a sua finalidade, deixando os morfismos para uma segunda fase.

#### 4.3.1. APLICAÇÃO DE MODELOS

Apesar da arquitectura do desenvolvimento orientado pelos modelos estar bem definida, a aplicação de diversos tipos de informação (e.g. formatos e artefactos visuais) à mesma pode não ser totalmente trivial. Para tal, nesta secção, vai-se descrever como se aplica os vários tipos de informação presentes na solução aos vários níveis presentes na arquitectura de modelação do desenvolvimento orientado pelos modelos.

##### 4.3.1.1. FORMATOS DE DADOS

De forma a se abstrair cada formato existente, os modelos entram para descrever a especificação dos formatos. Esta aplicação passa pelos quatro níveis em que o desenvolvimento opera. O primeiro nível, M0, é o nível dos dados representados por um formato específico. Mas estes dados, de forma a poderem ser utilizados em toda a arquitectura, necessitam de estar descritos numa forma onde se possam aplicar morfismos entre as diversas partes da arquitectura. Para tal, nos níveis M1, M2 e M3, vai ser descrita, de forma formal, qual e como a informação é utilizada por cada formato.

O modelo no nível M1 define especificamente como é que a informação pode ser descrita nos dados, sendo que ele próprio é definido de acordo com uma especificidade (i.e. qual os artefactos de modelação que podem ser utilizados, e.g. OMG UML), definida no meta-

---



modelo, o nível M2. O ultimo nível, o M3, descreve quais os artefactos que podem ser utilizados na criação dos meta-modelos (e.g. OMG MOF), contando com uma característica distinta, já atrás mencionada, que é o facto de serem modelos reflexivos, ou seja, descrevem-se a eles próprios, não necessitando de ter outro nível por cima a defini-los.

Na figura seguinte pode-se verificar a aplicação dos quatro níveis mencionados, ao caso específico de um formato particular.

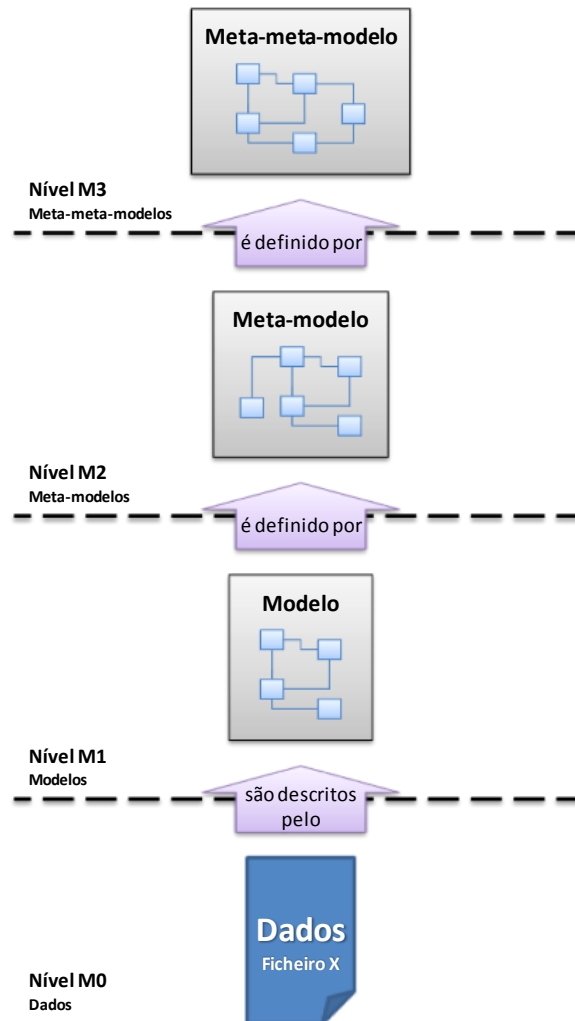


FIGURA 4.4 – FORMATO ESPECÍFICO ORIENTADO PELOS MODELOS.

#### 4.3.1.2. ARTEFACTOS VISUAIS

A aplicação dos artefactos visuais, já discutida anteriormente na secção 3.2 (página 55), centra-se num paradigma em que os artefactos para a visualização são vistos como modelos, de forma a permitir mais tarde os morfismos entre eles. Por exemplo, um grafo pode ser transformado num modelo, sendo definidos os seus nós e ligações que mais tarde podem ser aplicados na instanciação.

Com esta aproximação, pode-se aproveitar uma das grandes potencialidades do desenvolvimento orientado pelos modelos que é a criação das visualizações pelas pessoas que necessitam das mesmas e as vão utilizar, sem necessitarem de contar com conhecimentos técnicos de implementação. Esta aproximação está representada na figura seguinte.

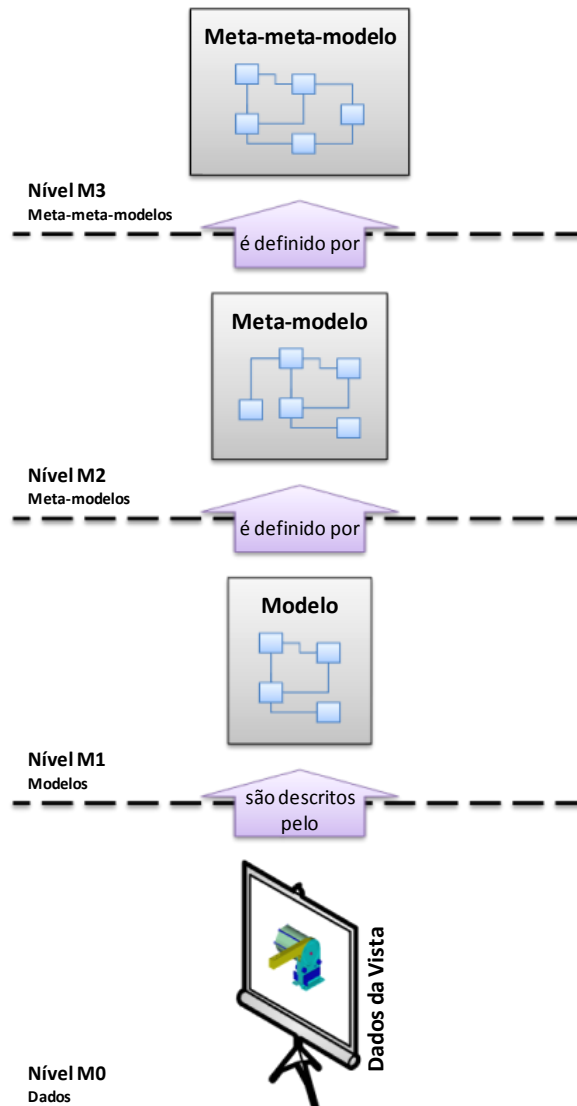


FIGURA 4.5 – VISTAS ORIENTADAS PELOS MODELOS.

#### 4.3.2. MORFISMOS DE MODELOS

Depois da aplicação dos modelos às várias componentes, é necessário passar à ligação entre elas recorrendo aos morfismos. Estes morfismos irão todos centrar-se na base comum referida anteriormente e portanto é dela que todos os morfismos irão partir.

## 4.3.2.1. HARMONIZAÇÃO DE FORMATOS

Começando pelos formatos específicos, os modelos aí definidos necessitam de ter um mapeamento com o da base comum. Este mapeamento, de forma a poder ser efectuado automaticamente utilizando ferramentas de desenvolvimento orientado pelos modelos, necessita que ambas as partes derivem do mesmo meta-meta-modelo.

Atrás foi referido que por vezes o mapeamento pode ser definido em dois níveis, dos meta-modelos: mapeamento de tipos, e dos modelos: mapeamento de instâncias. Que corresponde ao que acontece neste caso, devido ao facto de, por exemplo, um formato utilizar uma linguagem de modelação diferente para o descrever. Logo, é necessário aplicar o mapeamento ao nível da linguagem de modelação e dos formatos.

Estes mapeamentos, sempre que possível, devem ser bidireccionais de forma, a mais tarde permitir que se efectue tanto a importação como a exportação de dados para diversos formatos, como é demonstrado na figura seguinte.

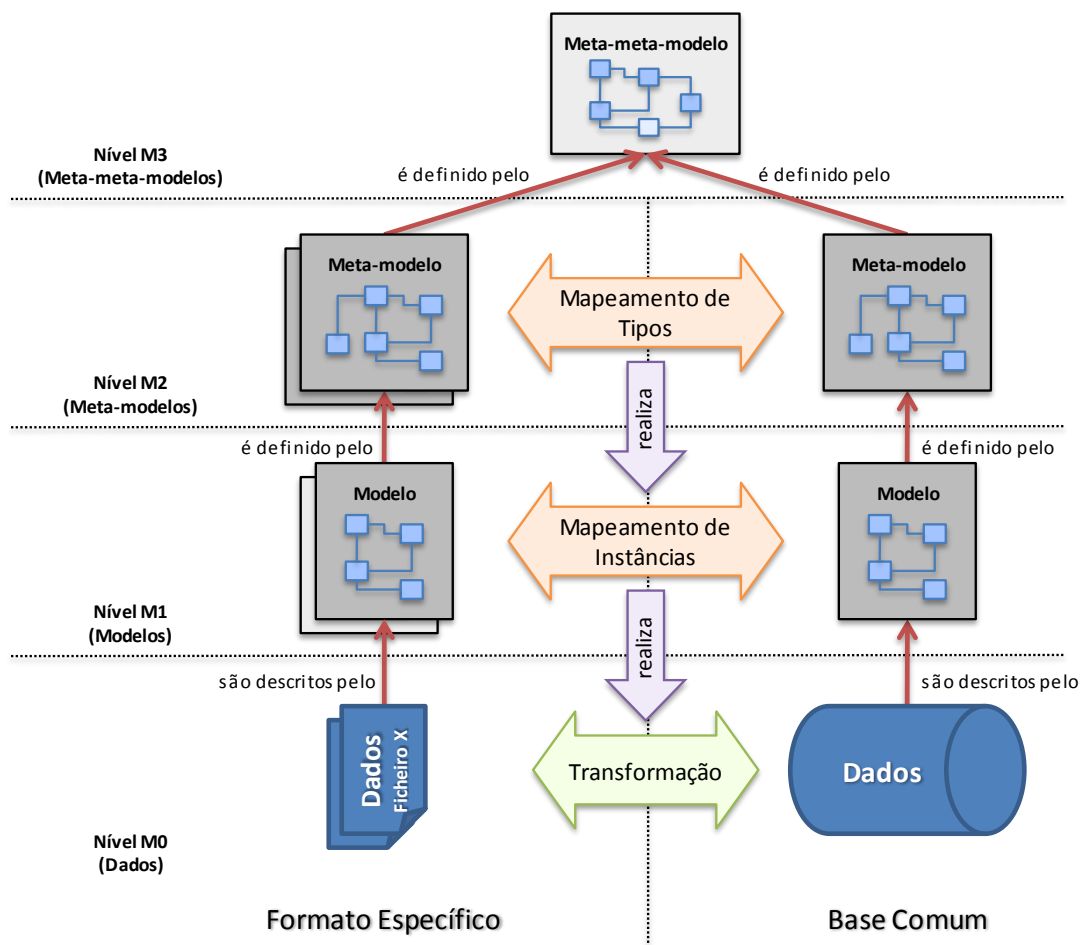


FIGURA 4.6 – MORFISMO ENTRE O FORMATO ESPECÍFICO E A BASE COMUM.

Importar os dados dos ficheiros para uma base comum permite a exportação para qualquer outro formato considerado na implementação da arquitectura. Mas este não é o único objectivo proposto. O próximo morfismo vai-se focar na aplicação de diferentes formas de visualização dos dados importados, logo é também realizado entre a base comum e as diversas vistas definidas.

#### 4.3.2.2. APLICAÇÃO DE ARTEFACTOS VISUAIS

No caso das vistas, tanto os mapeamentos como a transformação ocorrem de forma semelhante, sendo as novas vistas criadas com base na informação desejada. De notar que a operação é novamente definida bidireccionalmente, isto para se permitir que alterações efectuadas às vistas se possam reflectir nos dados. Pegando no exemplo anterior, utilizar a tabela para alterar os dados. Esta operação é demonstrada na figura seguinte.

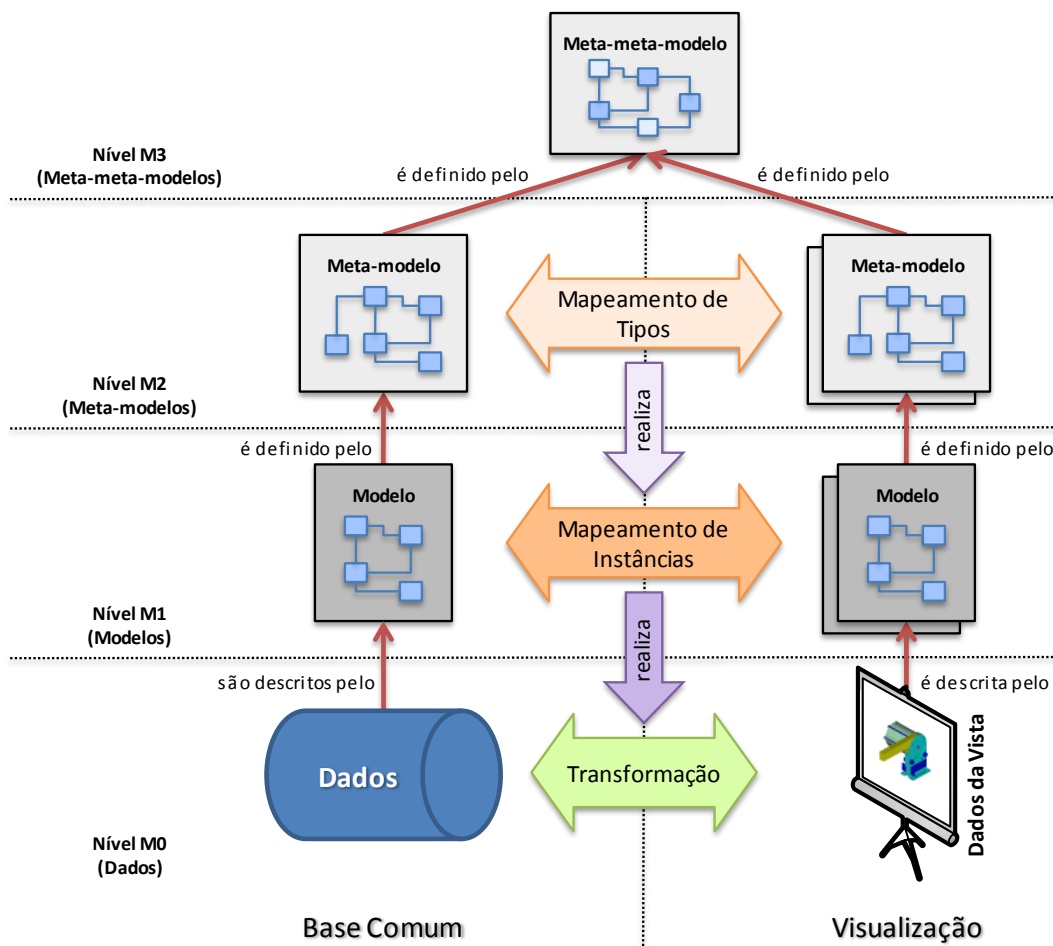


FIGURA 4.7 – MORFISMO ENTRE A BASE COMUM E A VISUALIZAÇÃO.

#### 4.3.2.3. CONTEXTUALIZAÇÃO DA INFORMAÇÃO

A contextualização da informação conta com uma abordagem diferente das efectuadas anteriormente. A grande diferença situa-se no mapeamento e na transformação, visto esta não ser efectuada entre dois modelos diferentes, mas sim directamente no modelo da base comum, ou seja, a contextualização é aplicada ao modelo da base comum, alterando-o de acordo com o contexto de aplicação.

Voltando então à transformação, sendo esta definida da forma atrás referida, resta aplicá-la aos dados da base comum, obtendo-se a informação filtrada de acordo com o pretendido como se pode verificar na figura seguinte.

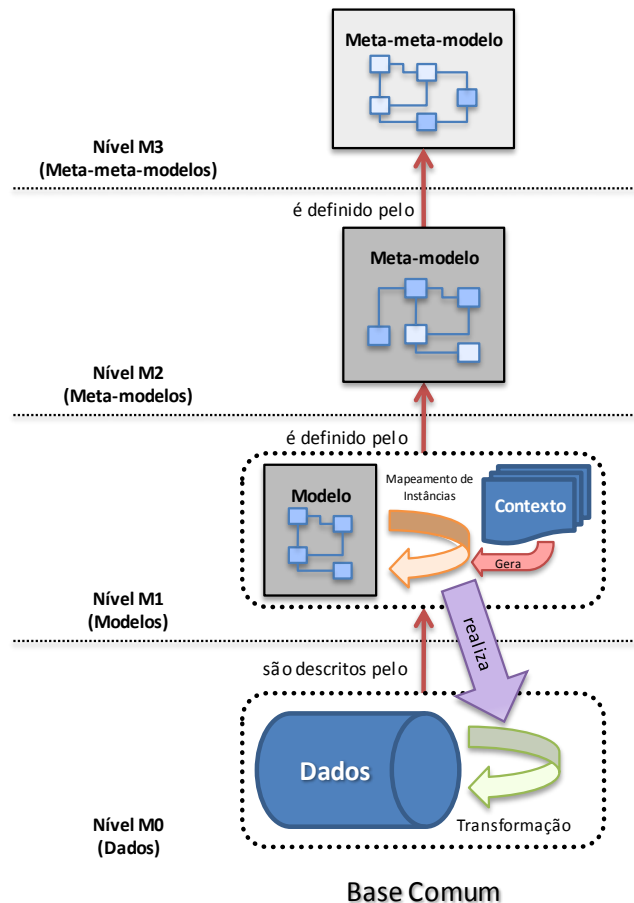


FIGURA 4.8 – MORFISMOS NA APLICAÇÃO DAS REGRAS DE CONTEXTUALIZAÇÃO DA INFORMAÇÃO.

#### 4.3.3. ARQUITECTURA GERAL

A arquitectura final para solucionar os requisitos previamente definidos é uma junção das várias partes apresentadas durante este capítulo, unidas pela base comum estabelecida previamente. O que resulta no diagrama apresentado na página seguinte.

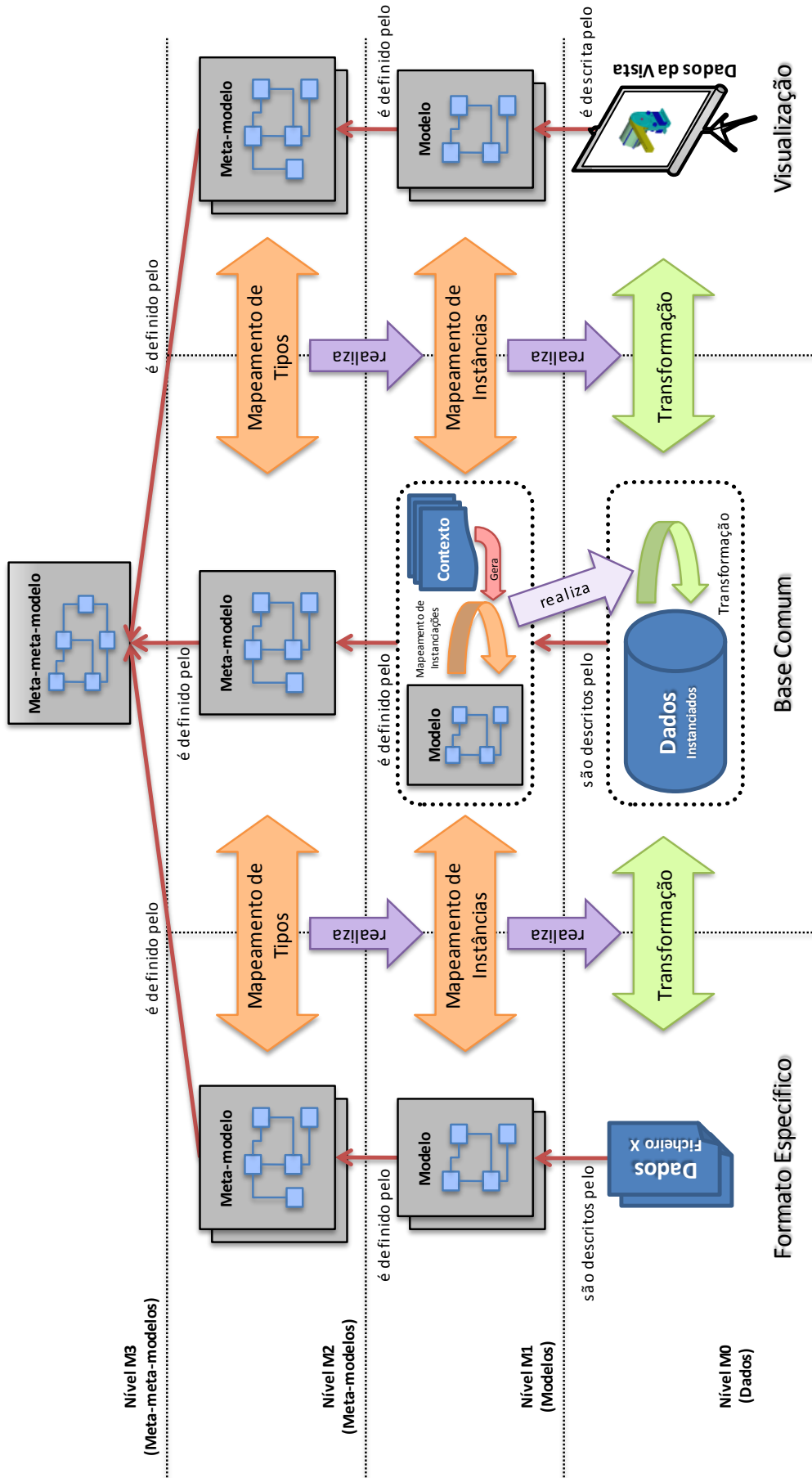


FIGURA 4.9 – ARQUITECTURA PARA A UNIFICAÇÃO DA VISUALIZAÇÃO DE DADOS DE PRODUTO.

## 5. IMPLEMENTAÇÃO DA SOLUÇÃO

De forma a se poder efectuar testes para a validação do conceito da solução proposta face aos desafios encontrados, é necessário proceder-se à criação de uma implementação que possa ser interposta num processo de validação através de experimentação.

Esta implementação enquadra-se no terceiro nível (Prova de conceito, demonstrada analiticamente e/ou experimentalmente) dos níveis de prontidão de tecnologia (Technology Readiness Levels). Este estipula, que esta etapa inclui tanto o estudo analítico para a definição das tecnologias apropriadas a serem utilizadas, como testes em ambiente controlado de forma a se validar o conceito da solução encontrada.

Estes estudos e testes são apresentados sem ter como objectivo ser uma implementação funcional com todas as capacidades próprias de um produto comercial. É, portanto necessário analisar os testes de forma a se perceber as potencialidades da solução proposta e não o desempenho específico da implementação efectuada. Logo é preciso levantar quais as falhas e melhoramentos para estes ficarem documentados, o que no caso de uma implementação comercial ajudaria a corrigir alguns problemas de raiz. (Mankins, 1995).

De seguida é efectuado o levantamento das tecnologias utilizadas na implementação da solução bem como a apresentação de como estas foram aplicadas a realização da prova de conceito. Os testes em ambiente controlado são estudados no capítulo seguinte.

### 5.1. ESCOLHAS TECNOLÓGICAS

Existem diversas decisões acerca das tecnologias a serem utilizadas que têm de ser tomadas:

1. Qual a linguagem de programação sob a qual a prova de conceito vai assentar;
  2. Que estrutura de dados se vai utilizar na base comum;
  3. Quais os formatos de representação geométrica que irão contar com a sua implementação;
  4. Quais as diferentes vistas escolhidas.
-

### 5.1.1. LINGUAGEM DE PROGRAMAÇÃO

O primeiro passo na escolha das tecnologias utilizadas para a prova de conceito é qual a linguagem de programação que vai suportar a implementação. Esta escolha recaiu sobre o Java em detrimento de outras devido essencialmente aos seguintes factos.

Logo porque se está perante uma linguagem que pode ser utilizada gratuitamente, e tem disponíveis inúmeras ferramentas, também de domínio público, para se poder trabalhar sobre ela, nomeadamente um dos IDE (Integrated Development Environment, em português, ambiente integrado para o desenvolvimento) mais conhecidos para desenvolvimento, o Eclipse<sup>5</sup>.

Também, o facto de a linguagem correr sobre uma máquina virtual (JVM – Java Virtual Machine, em português, Máquina Virtual do Java), permitindo o desenvolvimento de uma aplicação que é independente do sistema operativo (Sun Microsystems, Inc.).

Mais, e também importante, o facto de ser uma linguagem em que o autor tem experiência, promovendo uma mais facilidade e melhor qualidade na implementação da solução.

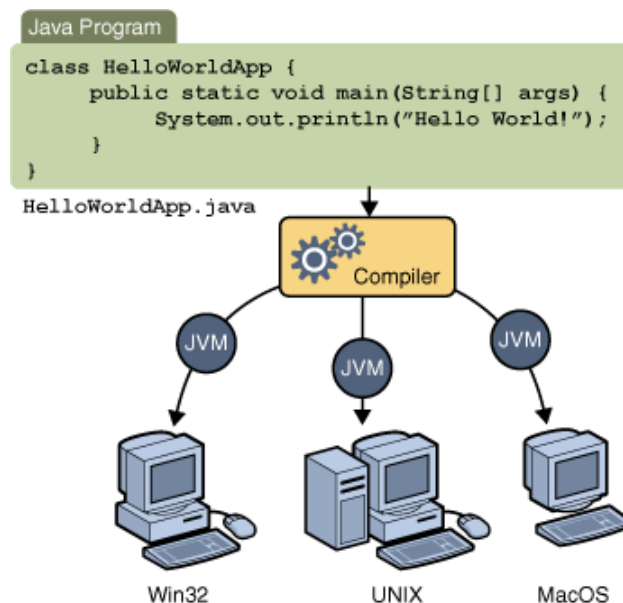


FIGURA 5.1 – FUNCIONAMENTO DA MÁQUINA VIRTUAL DO JAVA (SUN MICROSYSTEMS, INC.).

<sup>5</sup> Eclipse – Uma plataforma livre para o desenvolvimento – <http://www.eclipse.org/>



### 5.1.2. ESTRUTURA DE DADOS DA BASE COMUM

De seguida é preciso definir a tecnologia utilizada no elemento central desta implementação, a base comum. Esta necessita ter uma estrutura de dados com grande expressividade, suficiente para poder ser utilizada com o maior número de tipo de dados geométricos existentes nos diversos formatos actuais.

Anteriormente no capítulo 2.2 – Normas de troca de dados para a visualização, foi feito um levantamento de alguns dos formatos de maior relevo para a troca de dados geométricos. Como o objectivo de uma implementação do tipo prova de conceito é apenas um teste funcional à solução proposta, não se justifica desenvolver uma nova estrutura de dados para descrever todos os artefactos geométricos passíveis de ser utilizados nos formatos, então foi tomada a decisão de escolher um dos formatos que conta com a maior capacidade de descrição e utilizá-lo como base comum.

Analisando portanto a Tabela 2.1 – Características das várias normas de representação geométrica (página 43), pode-se verificar que, dos formatos analisados, existem quatro que conseguem descrever todos os tipos de informação geométrica analisados. Desses quatro a escolha recai sobre o X3D, porque, além de ter sido o formato com melhor desempenho nos critérios definidos para a troca de dados geométricos orientado à visualização, é o único dos quatro que é normalizado por uma entidade internacional, neste caso a ISO.

O X3D tem definido de forma abstracta toda a sua descrição dos dados através da norma ISO/IEC FDIS 19775-1.2:2008. Esta descreve artefactos geométricos como modelos bidimensionais, tridimensionais, animações, interacção com o utilizador, navegação, simulação física, CAD, sistemas de partículas, etc. Visto este descrever um grande tipo de artefactos geométricos, é um formato com uma grande expressividade, o que se adequa às necessidades da base comum.

Portanto, de forma a se poder aplicar o X3D à arquitectura orientada pelos modelos é ainda necessário que esteja descrito numa linguagem de modelação. Neste caso, existe uma outra norma, a ISO/IEC FCD 19776-1.2:200x (Ainda não publicada, mas é a versão correspondente à norma atrás apresentada, ao contrario da já publicada e desactualizada, ISO/IEC

---

19776:2005) que descreve o relacionamento entre a abstracção definida, e a linguagem XSD<sup>6</sup>, sendo disponibilizado um esquema completo da norma atrás referida em XSD.

Estando o modelo do X3D definido em XSD, este entra portanto no nível dos modelos, M1. No nível M0, os dados, entram os dados descritos em X3D, tipicamente um ficheiro XML de acordo com o XSD do X3D. Ao nível M2, dos meta-modelos, está a definição do XSD, normalizado por três recomendações do consórcio para a *World Wide Web* (W3C): “XML Schema Part 0: Primer Second Edition”, “XML Schema Part 1: Structures Second Edition” e “XML Schema Part 2: Datatypes Second Edition”. Estas definições são descritas utilizando a linguagem de modelação MOF, que se encontra ao nível dos meta-meta-modelos M3.

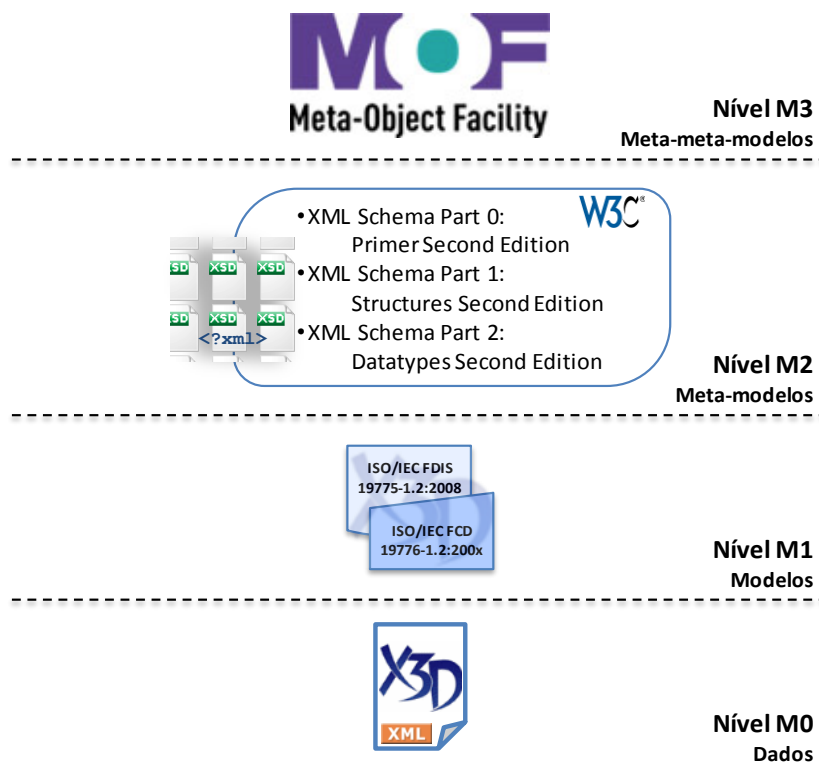


FIGURA 5.2 – X3D APLICADO AO DESENVOLVIMENTO ORIENTADO PELOS MODELOS.

### 5.1.3. FORMATOS GEOMÉTRICOS

Depois de definida a base comum, é necessário quais os formatos que irão contar com a sua presença na implementação. O primeiro, e óbvio, o X3D, visto a base comum ser descrita através dele. Este conta com a sua implementação facilitada, pois ao ser o mesmo da base comum, os morfismos a aplicar são directos.

<sup>6</sup> XSD –Definição dos esquemas de XML – <http://www.w3.org/2001/XMLSchema>

A outra escolha recai sobre o formato promovido pela AutoDesk, o DWG. A razão desta escolha advém do facto de a maior parte dos ficheiros de dados trocados no caso de estudo deste trabalho se encontrarem descritos neste formato. Este conta com algumas adversidades na sua implementação, principalmente por não ter disponível publicamente a sua especificação.

Este problema é parcialmente resolvido com a especificação publicada de forma gratuita pela, já atrás referida, “Open Design Alliance”. Mas esta especificação apenas conta com a definição da estrutura de dados necessária para a aproximação orientada aos modelos, e com a forma de se interpretar os ficheiros, não existindo informação de como voltar a gravá-los, logo a implementação vai ficar limitada apenas à leitura do DWG, e não leitura e escrita como seria pretendido. Outra barreira existente na aplicação do DWG ao desenvolvimento orientado pelos modelos, é o facto de este não se encontrar descrito em nenhuma linguagem de modelação, para tal, para esta inclusão ser possível é necessário a transcrição de toda a sua especificação para uma linguagem de modelação adequada, e.g. UML.

#### 5.1.4. VISTAS

Falta portanto definir as formas de visualização que irão ser utilizadas. Neste caso foram escolhidas duas, a primeira consiste num visualizador tridimensional, sendo escolhido o Xj3D<sup>7</sup>, que é um visualizador que corre em ambiente Java. Este tem uma característica de relevo para a utilização nesta arquitectura, já que utiliza a mesma estrutura de dados definida na base comum, facilitando assim em grande medida a implementação, além de ter disponível essa estrutura, já descrita numa linguagem de modelação adequada, o UML.

A outra visualização definida foi uma vista tabular, utilizada para mostrar a equivalência entre as cores apresentadas nos modelos e os materiais que as constituem, contando com a possibilidade de se definirem texturas para cada material, que se reflectem na visualização tridimensional. Esta vista também necessita da definição do seu meta-modelo. Logo escolhendo-se o UML como linguagem de modelação, o seu meta-modelo resulta no apresentado na figura seguinte.

---

<sup>7</sup> Xj3D – Visualizador de cenas X3D - <http://www.xj3d.org/>

---

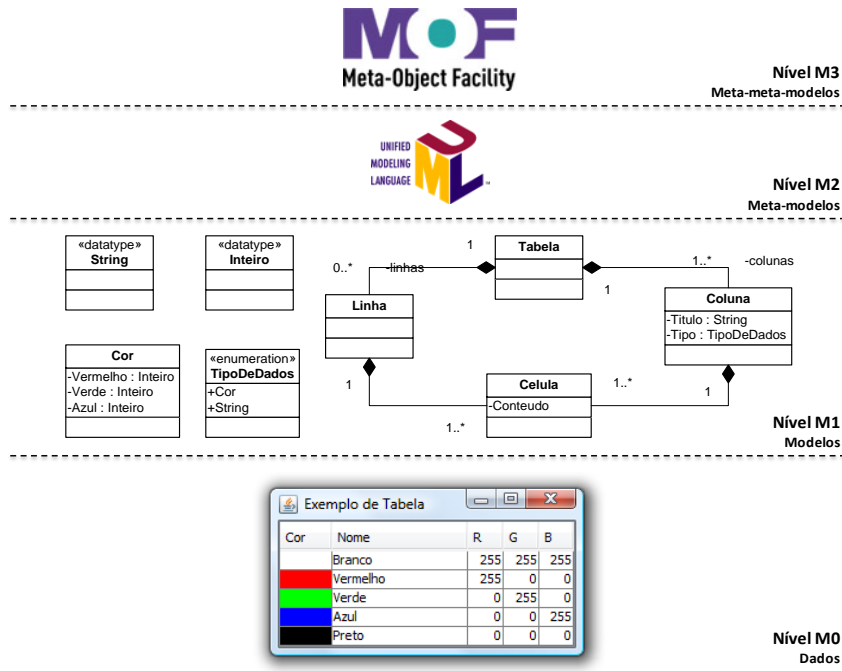


FIGURA 5.3 – TABELA DESCRITA EM UML.

### 5.2. INSTANCIÇÃO TECNOLÓGICA À ARQUITECTURA

A aplicação das escolhas efectuadas à arquitectura resulta na definição de linguagens de modelação, modelos, e tipos dados. Na figura seguinte, é apresentado um exemplo, de como estas escolhas tecnológicas (DWG, como formatos específico, X3D, como base comum, e Xj3D como visualizador) se enquadram na arquitectura proposta.

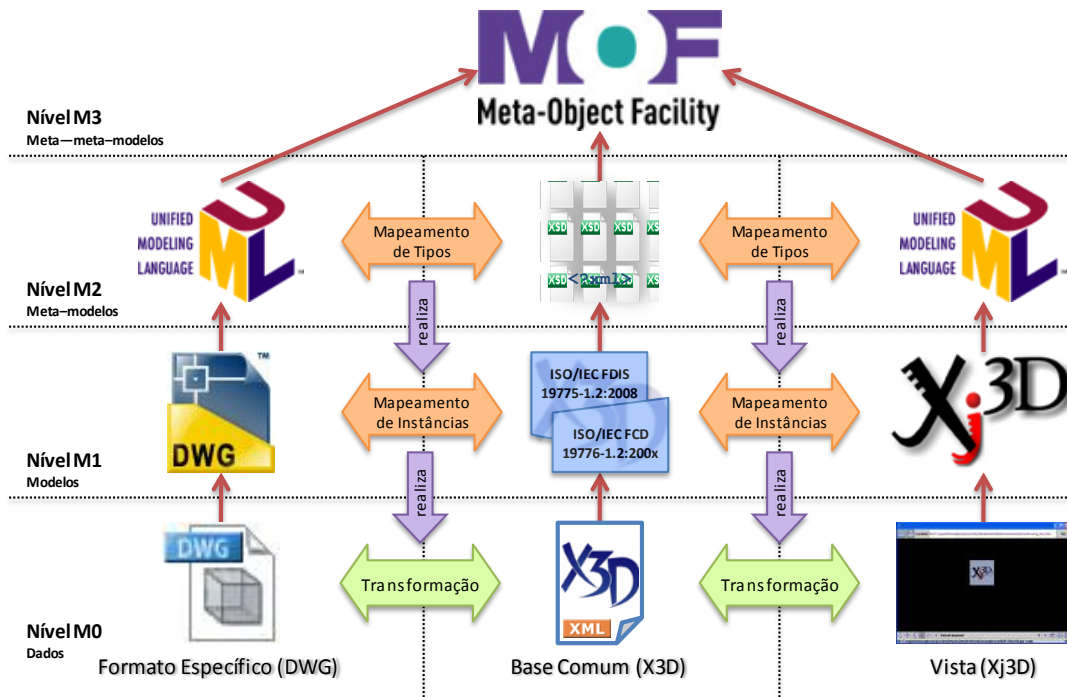


FIGURA 5.4 – APLICAÇÃO DAS ESCOLHAS TECNOLÓGICAS À ARQUITECTURA PROPOSTA.

### 5.3. PROCESSO DE IMPLEMENTAÇÃO

Estando escolhidas as tecnologias a utilizar, vai-se proceder à aplicação das mesmas. Esta aplicação vai passar por quatro passos, que concebem a implementação da solução:

1. Implementação da base comum;
2. Ligação da base comum aos formatos geométricos;
3. Ligação da base comum com as vistas;
4. Aplicação da contextualização da informação.

#### 5.3.1. BASE COMUM

Começando-se pela base comum, ao analisar a norma que define o X3D, verifica-se que esta é extensa, portanto o desenvolvimento orientado por modelos contribui para a automatização da sua implementação. Portanto o caminho a seguir é encontrar uma forma de automatizar o processo de transformação da norma do X3D em classes de java.

Na comunidade existem diversas ferramentas para efectuar a transformação de esquemas XML (XSD) para classes java. A escolha recaiu sobre o JAXB<sup>8</sup>, devido à sua simplicidade de utilização e da sua API estar integrada nas distribuições do java.

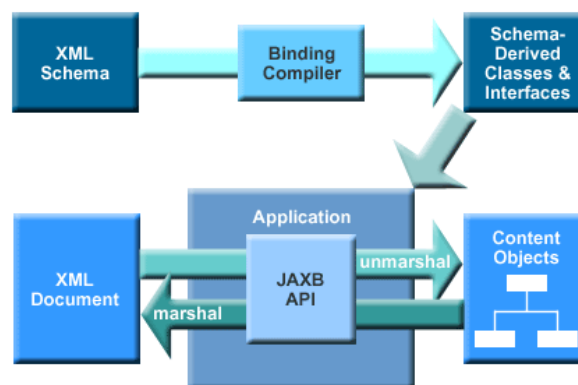


FIGURA 5.5 – ARQUITECTURA DO JAXB.

Utilizando esta abordagem obtém-se em classes de java toda a descrição da norma do X3D sem a necessidade de se recorrer à codificação manual da mesma. Resultando esta transformação em toda uma hierarquia de classes em memória que serão utilizadas no

<sup>8</sup> JAXB – Java Architecture for XML Binding – Arquitectura java para ligação do XML – <https://jaxb.dev.java.net/>

desenvolvimento da prova de conceito. De seguida é apresentado um exemplo de como um bloco da descrição em XSD resulta em código java.

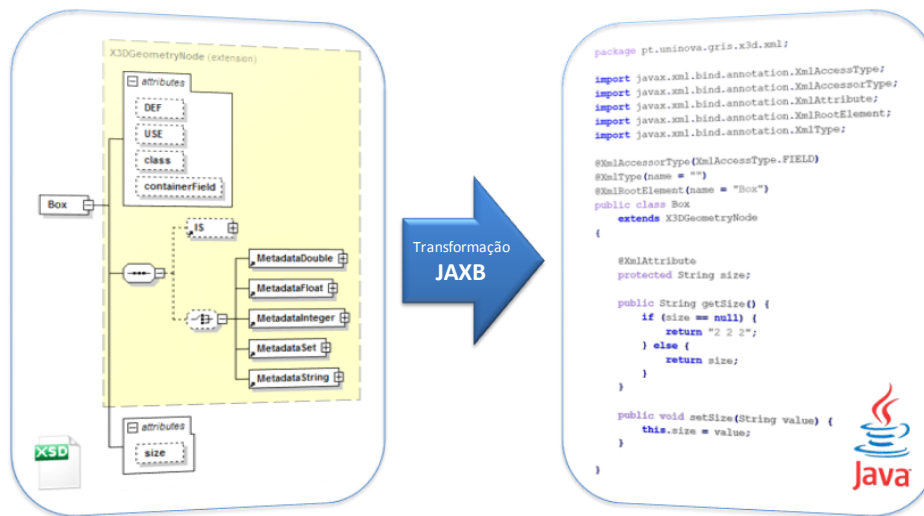


FIGURA 5.6 – EXEMPLO DE TRANSFORMAÇÃO OBTIDA COM A UTILIZAÇÃO DO JAXB.

### 5.3.2. LIGAÇÃO BASE COMUM COM FORMATOS GEOMÉTRICOS

Estando disponível a base comum em código Java, pode-se passar à ligação da mesma com os vários formatos. No caso do X3D, esta ligação é trivial, utilizando-se o *marshaller* e o *unmarshaller*, que consiste em utilizar a transformação atrás descrita para transformar os dados em memória em ficheiros XML correspondentes de acordo com a norma do X3D e vice-versa. No caso do DWG esta ligação não é trivial como no X3D, antes pelo contrário, é uma das partes mais exigentes a nível de implementação, a interpretação dos ficheiros DWG. Apesar de esta conceptualização foi efectuada com uma abordagem orientada por modelos, vai-se proceder a um atalho na mesma.

Este atalho vem do facto de a descrição do DWG não existir descrita formalmente, e por existir disponível online uma biblioteca chamada *jdwglib*<sup>9</sup> que contém, já em Java, as classes correspondentes à especificação do formato, logo leva a que não tenha sido efectuado o trabalho extenso de descrever o DWG numa linguagem de modelação. Neste caso esta abordagem é suficiente, pois apenas se está perante uma prova de conceito, e a grande contribuição do desenvolvimento orientado pelos modelos nesta fase seria a automatização da criação do código necessário.

<sup>9</sup> *jdwglib* – Biblioteca em java para o acesso a ficheiros DWG - <http://sourceforge.net/projects/jdwglib/>

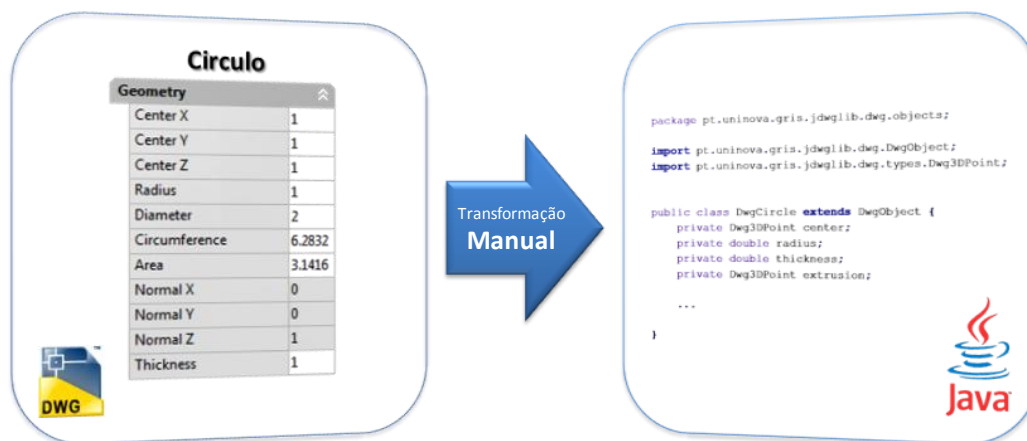


FIGURA 5.7 – EXEMPLO DE TRANSFORMAÇÃO DE OBJECTO DWG EM CLASSES JAVA.

Resistem no entanto dois problemas associados a esta escolha. O primeiro a um nível mais técnico, o facto de esta biblioteca suportar apenas a versão 2000 do formato (a maioria dos ficheiros de teste disponíveis são da versão 2004), sendo solucionado, com a implementação de acordo com a especificação disponibilizada pela Open design Alliance da versão 2004 (versão mais recente disponível aquando a implementação). Esta versão conta com vários melhoramentos, sendo os de relevo a inclusão de métodos de encriptação e compressão.

O outro problema vem do afastamento à abordagem orientada pelos modelos, impossibilitando que o mapeamento entre os formatos seja definido numa linguagem de transformação, tendo de ser definido directamente no código da implementação. Apesar dos morfismos entre o DWG e a base comum em X3D serem definidos directamente no código, os mesmos regem-se pela abordagem definida pelos modelos, sendo só aplicados ao nível das instâncias. No ANEXO B (página 115) podem ser encontrados em pormenor os morfismos definidos.

### 5.3.3. LIGAÇÃO BASE COMUM ÀS VISTAS

Para se utilizar a solução na sua totalidade falta aplicar visualizações aos dados descritos na base comum. Como já referido anteriormente, os dados utilizados pelo visualizador tridimensional regem-se pelo mesmo modelo de dados. Logo a implementação deste é apenas a utilização da biblioteca disponibilizada com os dados descritos na base comum.

No caso da vista tabular, que apresenta uma listagem das cores e materiais que lhe estão associados, é utilizada a mesma abordagem da contextualização da informação (o agrupamento de objectos por cor) extraindo-se a meta-informação que descreve os

materiais utilizados. Outra funcionalidade interessante é a possibilidade de se escolher uma imagem para a textura, cuja escolha é reflectida no visualizador tridimensional.

#### 5.3.4. APLICAÇÃO DA CONTEXTUALIZAÇÃO DA INFORMAÇÃO

Sendo então possível a interpretação de dois formatos para a base comum, a exportação dos dados em X3D e a aplicação de vistas aos mesmos fica a faltar a possibilidade de contextualizar essa informação. Esta contextualização, tal como definido na arquitectura para a solução, vai ser efectuada na base comum.

Como neste trabalho, o grande foco é a harmonização dos formatos e a visualização, a contextualização da informação vai ser efectuada implicitamente no código. Esta opção foi tomada, porque apenas se quer testar a sua aplicabilidade, e não desenvolver toda uma estrutura onde se possam representar as regras de contextualização. Nesta implementação a única regra definida foi o agrupamento dos objectos da mesma cor em grupos, visto, no caso de uso desta implementação, a cor ter uma relação directa com o material utilizado.

#### 5.4. INTERFACE DE UTILIZAÇÃO

Esta implementação para estar finalizada, necessita ter uma forma de os utilizadores utilizarem todos o trabalho atrás efectuado. Para tal foi definida uma interface de utilização que vai de encontro às funcionalidades já definidas, permitindo abrir ficheiros em DWG, exportar os ficheiros para X3D com ou sem regras de contextualização, e ter disponível as duas vistas definidas. Esta interface resultou na aplicação observada na figura seguinte.

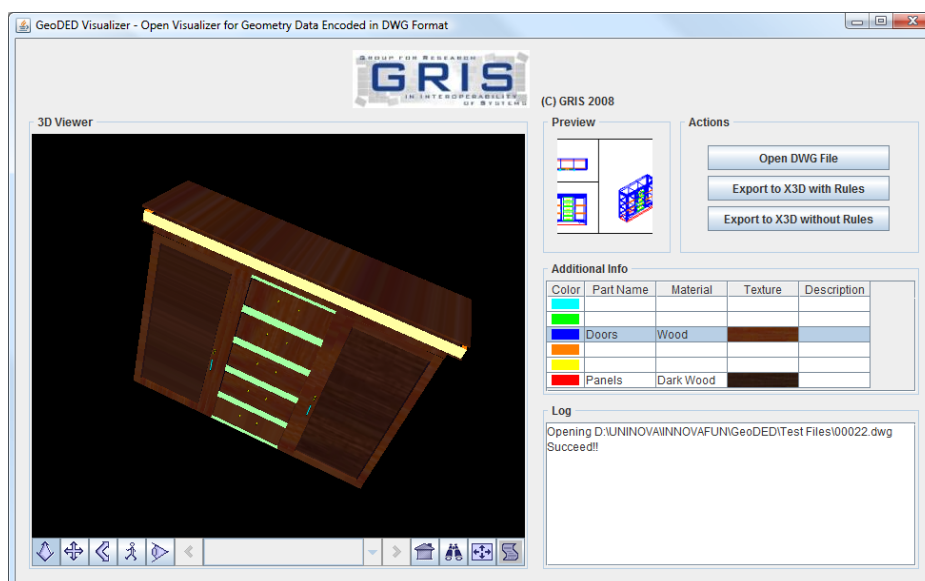


FIGURA 5.8 – INTERFACE GRÁFICA DA APLICAÇÃO DESENVOLVIDA PARA A PROVA DE CONCEITO.



## 6. TESTES E VALIDAÇÃO

O principal objectivo desta secção é o efectuar de testes funcionais de forma a se aferir o cumprimento do sistema face aos objectivos que foram definidos inicialmente. Testar é o processo de procurar erros na implementação de um sistema através de experiencias. Sendo que as mesmas são executadas num ambiente controlado onde se simula o uso real do sistema. Portanto o objectivo dos testes é o ganhar de confiança que caso o sistema seja utilizado num ambiente não controlado ele vai funcionar de forma satisfatória. Os testes podem apenas provar a presença de erros, mas um teste com sucesso não implica que o sistema está livre de erros (Tretmans, 2001).

De forma a se efectuar então a validação da conceptualização efectuada, é necessário proceder-se à definição, tanto de uma metodologia de testes, como de um conjunto de testes a serem efectuados sobre a prova de conceito implementada de acordo com a mesma. Como a solução apresentada passa por três aspectos integrados numa só solução, são necessários definir testes funcionais a cada uma destas vertentes. De seguida são apresentados, tanto a metodologia de testes adoptada, como a definição de quais os testes efectuados e os seus resultados, rematando-se com uma conclusão final sobre a aptidão da solução para aferir os cumprimentos definidos.

### 6.1. METODOLOGIA PARA TESTES

Existem diversas metodologias para testar a aptidão de soluções para o cumprimento dos seus requisitos, tendo cada uma o seu domínio específico de aplicação (Onofre, 2007). Apesar de não serem todas orientadas para o mesmo fim, algumas, têm semelhanças, sendo a semelhança que mais se evidencia a utilização da norma ISO/IEC 9646 como ponto de partida. Como tal, e visto que se quer uma metodologia de testes abstracta, vai-se utilizar os conceitos definidos por essa norma.

#### 6.1.1. ISO/IEC 9646 – PLATAFORMA E METODOLOGIA PARA TESTES DE CONFORMIDADE

Esta norma foi originalmente desenvolvida para disponibilizar uma plataforma e definir uma terminologia para a aplicação de testes em sistemas OSI (“Open System Interconnection”, Interconexão em sistemas abertos). Mas devido ao seu pouco uso, a metodologia tem sido pouco utilizada para testes de conformidade a estes sistemas. Não obstante, a metodologia

---

tem sido aplicada a outros tipos de protocolos e sistemas, sendo utilizada como base para outras metodologias de testes de conformidade, como a utilizada na norma ISO 10303 (i.e. “ISO 10303 part 30 – Conformance testing methodology and framework”).

O processo de testes descrito por esta metodologia é dividido em três fases (evidenciadas na Figura 6.1). A primeira fase é a especificação de um conjunto de testes abstractos para o sistema em questão, e é referida como definição de testes. Os testes são abstractos no sentido em que são desenvolvidos independentemente de qualquer implementação. A segunda fase consiste na definição dos testes de forma a poderem ser executados, e denomina-se implementação de testes. Nesta fase passa-se a ter em conta a implementação que vai ser testada, adaptando-se os testes definidos anteriormente à implementação do sistema. A última fase, a execução de testes, consiste na sua execução e observação dos resultados. O que leva a um veredicto sobre a conformidade do sistema em teste com os requisitos definidos inicialmente (Tretmans, 2001).

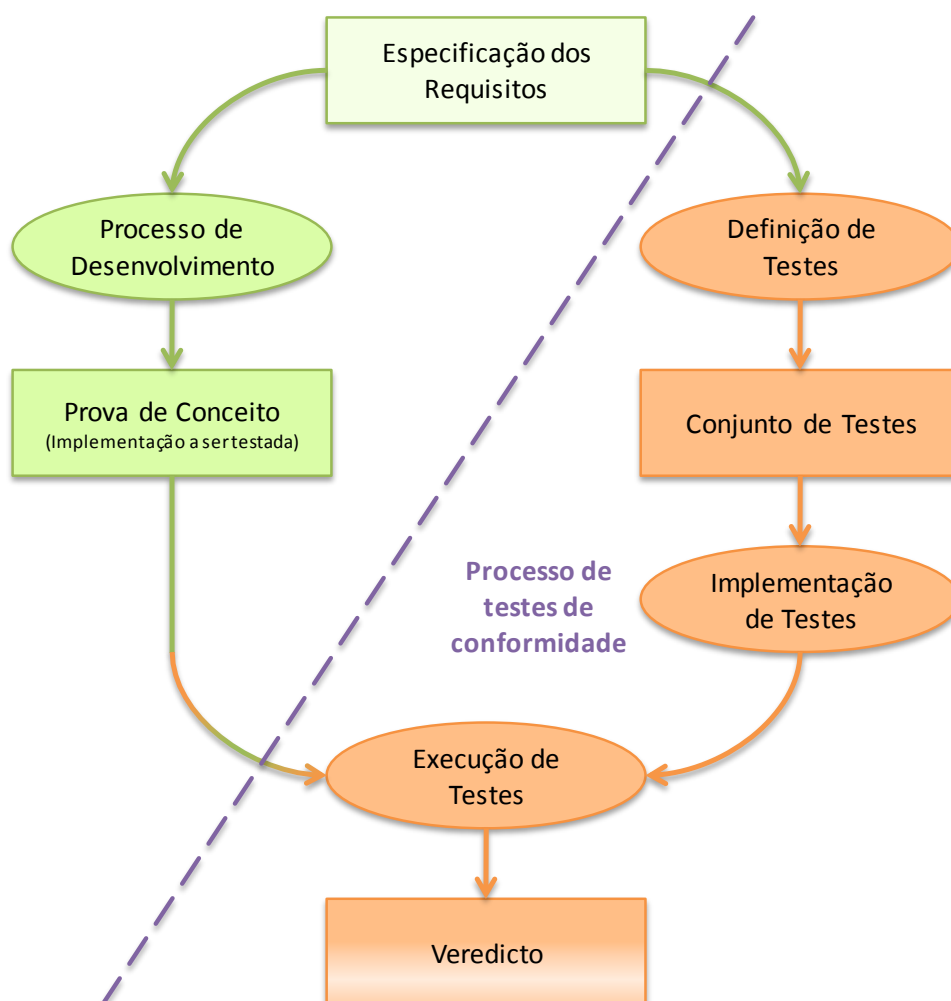


FIGURA 6.1 – VISTA GLOBAL DO PROCESSO DE TESTE DE CONFORMIDADE.

### 6.1.2. NOTAÇÃO PARA TESTES – TTCN – NOTAÇÃO TABULAR E DE ÁRVORE COMBINADA

Como a metodologia de testes é normalizada, ela deve ser especificada com uma notação bem definida, independente e aceite globalmente. A norma ISO/IEC 9646 recomenda a utilização da linguagem semi-formal TTCN, “Tree and Tabular Combined Notation” – Notação Tabular e de Árvore Combinada.

Esta notação rege-se pelo modelo de caixa preta, i.e. onde o comportamento interno do sistema não é relevante, e apresenta-se sobre a forma tabular as várias partes que definem o teste, sendo elas a vista geral sobre o teste, as declarações necessárias para a execução, as restrições e a parte dinâmica. É nesta última, na parte dinâmica, que os testes em si são descritos, i.e. o seu comportamento.

No TTCN o comportamento dos testes é definido pela sequência de eventos que ocorrem durante o teste. Uma sequência pode ter diversas alternativas onde diversos comportamentos podem ser levados em conta, dependendo das diversas respostas do sistema aos pedidos efectuados. Este comportamento é definido em forma de árvore, e uma cadeia de sucessivos eventos é indicada aumentando-se a indentação dos vários eventos, sendo que os eventos alternativos são definidos utilizando a mesma indentação (i.e. evento a e seguir caso o presente ao mesmo nível falhe). Sendo que a sequência de eventos acaba com a definição de um veredicto, tanto que diferentes comportamentos podem tomar diferentes veredictos, e.g. pode existir mais do que um comportamento que corresponda ao sucesso do teste.

Os vários eventos são divididos em duas partes: acções e questões. As acções, representadas com um ponto de exclamação no início do evento, definem as interacções com o sistema, sendo que as questões, representadas por um ponto de interrogação, são as respostas esperadas do sistema. O veredicto pode ter três resultados diferentes, pode acabar com um veredicto verdadeiro, de sucesso, um veredicto de teste falhado, de falha, e um veredicto de inconclusivo que define que o teste foi correcto mas não com um comportamento desejado (Tretmans, 2001).

Mais informação acerca da utilização desta notação pode ser encontrado num tutorial online do IEC, (International Engineering Consortium, 2007), de onde foi retirado o exemplo apresentado de seguida, onde é descrito o comportamento de um teste para se verificar se uma chamada telefónica pode ser efectuada.

---

TABELA 6.1 – EXEMPLO DE TESTE DEFINIDO EM TTCN.

Teste	
<b>Nome do teste:</b> Conexão Básica	
<b>Grupo:</b>	
<b>Propósito:</b> Verificar se uma chamada telefónica pode ser estabelecida	
<b>Comentários:</b>	
Comportamento	Veredicto
! Levantar auscultador	
? Tom de marcação	
! Marcação do número	
? Tom de chamada	
? Linha conectada	
! Desligar auscultador	Sucesso
? Ocupado	
! Desligar auscultador	Inconclusivo
? Tom de marcação ausente	Falha

A descrição da parte comportamental do exemplo apresentado pode ser explicado da seguinte forma em linguagem textual:

1. O utilizador levanta o auscultador;
2. Testa se o tom de marcação é ouvido;
3. Se o tom de marcação for ouvido, então o utilizador marca o número;
4. Este verifica se o tom de chamada é ouvido;
5. Se o tom de chamada é ouvido, ele testa se a ligação pode ser estabelecida;
6. Se poder, ele desliga o auscultador e define o veredicto como sucesso;
7. Se o tom de chamada não for ouvido, mas sim o tom de ocupado;
8. O auscultador é desligado, e o veredicto é definido como inconclusivo;
9. Se o tom de marcação não for ouvido, então o veredicto é falha, e o teste é terminado.

## 6.2. TESTES FUNCIONAIS

Definida a metodologia a utilizar na realização dos testes pode-se proceder à definição dos testes de uso que irão resultar no veredicto sobre a aptidão da solução encontrada no cumprimento dos requisitos.

Neste caso, os testes vão ser efectuados manualmente, devido a duas razões. Primeiro a prova de conceito desenvolvida requerer a interacção humana no seu funcionamento,

segundo, a apresentação de resultados é efectuada de diversas formas, nomeadamente formas visuais que dificultam uma abordagem automática ao processo.

Os testes vão ser divididos em três grandes grupos, sendo cada um correspondente a cada requisito levantado neste trabalho, a harmonização dos formatos, a personalização de vistas e a contextualização da informação.

De seguida é apresentada uma sinopse dos testes definidos e dos seus resultados, sendo que o detalhe de execução de cada um é apresentado no ANEXO C (página 125).

#### 6.2.1. TESTES DA HARMONIZAÇÃO DOS FORMATOS

Como nesta implementação apenas foram definidos morfismos entre dois formatos, vai-se proceder apenas aos testes de transformação dos vários tipos de objectos contemplados de um formato para outro, neste caso objectos descritos em DWG versão 2004 para objectos descritos em X3D.

A análise de resultados começa com a passagem do ficheiro X3D resultante num validador de XML, neste caso o Altova XMLSpy. Depois passa-se para a comparação das características dos objectos descritos no DWG, levantadas recorrendo-se ao software AutoCAD da AutoDesk, e aos resultados obtidos no X3D, levantados recorrendo-se a um comum editor de texto, sendo que a comparação vai ser efectuada sem o recorrer de nenhum método automático, mas sim uma comparação manual entre as características levantadas. Também vai fazer parte da análise uma apreciação visual dos resultados, recorrendo-se neste caso ao visualizador desenvolvido pelo “Fraunhofer Institute for Computer Graphics (IGD-A4)”, InstantPlayer versão beta 5.

Depois de analisados os morfismos individualmente, vai ser apresentado um exemplo composto, i.e. um desenho real composto por diversos tipos básicos testados, onde a análise comparativa vai ser efectuada recorrendo-se somente à comparação visual de ambos os resultados.

##### 6.2.1.1. TESTE DE MORFISMO – ELEMENTOS GRÁFICOS SIMPLES

Este teste consiste em validar, individualmente a capacidade que os morfismos definidos têm de equivaler elementos gráficos simples descritos no DWG em X3D. Os detalhes deste

---

teste podem ser encontrados no ANEXO C, secção C.1.1 – Teste de morfismo – Elementos gráficos simples (Página 125), sendo apresentado de seguida um resumo da sua execução.

TABELA 6.2 – DETALHE DO TESTE DE MORFISMO – ELEMENTOS GRÁFICOS SIMPLES.

<b>Teste</b>	
<b>Nome do teste:</b>	Morfismo de elementos simples (Arco, Circulo, Linha, Face 3D e Sólido)
<b>Grupo:</b>	Harmonização dos formatos
<b>Propósito:</b>	Verificar o funcionamento do morfismo entre elementos geométricos simples
<b>Comentários:</b>	O ficheiro DWG utilizado vai ser da versão 2004
<b>Comportamento</b>	<b>Veredicto</b>
! Criar um desenho no AutoCAD com os elementos simples referidos	
! Gravar o desenho obtido no formato DWG versão 2004	
! Abrir o ficheiro obtido na implementação do sistema	
? Ficheiro aberto com sucesso	
! Exportar o ficheiro para X3D	
? Exportação efectuada com sucesso	
! Proceder à comparação entre original e exportado	
? Resultado corresponde ao desenhado	Sucesso
? Resultado diferente do desenhado	Falha
? Exportação falhada	Falha
? Ficheiro não abriu	Falha

Para este teste foi criado um ficheiro em DWG versão 2004 que contém todos os elementos simples cujos morfismos foram definidos neste trabalho. Sendo, que posteriormente este foi aberto na implementação do sistema e exportado com sucesso para X3D.

De forma, a se identificar se o teste foi um sucesso ou não, foram efectuadas três tipos de análise. Primeiramente constou que o ficheiro obtido é valido, quando interposto face ao esquema que define a sua validade. Uma segunda fase foi a análise visual do resultado face ao desenho original, que neste caso também se relevou um sucesso. Por último, foi realizada uma análise minuciosa de cada um dos elementos geométricos testados, análise que também resultou em sucesso, pois todos os elementos no X3D corresponderam aos originais. Portanto, pode-se concluir que o veredicto deste teste foi de sucesso.

#### 6.2.1.2. TESTE DE MORFISMO – DESENHO COMPOSTO

Este teste, análogo ao anterior, consiste em validar, a capacidade que os morfismos definidos têm de equivaler um exemplo real, que poderia ser utilizado numa qualquer troca industrial, descrito em DWG em X3D. Os detalhes deste teste podem ser encontrados no

ANEXO C, secção C.1.2 – Teste de morfismo – Desenho composto (Página 133), sendo apresentado de seguida um resumo da sua execução.

TABELA 6.3 – DETALHE DO TESTE DE MORFISMO – DESENHO COMPOSTO.

Teste	
<b>Nome do teste:</b>	Morfismo de um desenho composto
<b>Grupo:</b>	Harmonização dos formatos
<b>Propósito:</b>	Verificar o funcionamento do morfismo na aplicação a um desenho composto
<b>Comentários:</b>	O ficheiro DWG utilizado vai ser da versão 2004
Comportamento	Veredicto
! Criar um desenho composto no AutoCAD	
! Gravar o desenho obtido no formato DWG versão 2004	
! Abrir o ficheiro obtido na implementação do sistema	
? Ficheiro aberto com sucesso	
! Exportar o ficheiro para X3D	
? Exportação efectuada com sucesso	
! Proceder à comparação entre original e exportado	
? Resultado corresponde ao desenho	Sucesso
? Resultado diferente do desenho	Falha
? Exportação falhada	Falha
? Ficheiro não abriu	Falha

Este teste, foi efectuado de forma análoga ao anterior, mas recorrendo-se a um desenho já existente, que corresponde a um desenho real, neste caso uma representação em CAD de uma mesa. Sendo que neste caso foi efectuada apenas uma comparação visual, devido à complexidade do desenho envolvido.

Analisando portanto o resultado da exportação visualmente, pode-se concluir que o desenho corresponde ao original, levando que o veredicto final em relação à harmonização de formatos dite que a solução obtém os resultados pretendidos, uma conversão de formatos efectuada com sucesso.

#### 6.2.2. TESTES DA PERSONALIZAÇÃO DE VISTAS

Tendo sido utilizadas duas vistas distintas nesta implementação, ambas vão ser alvo de testes e validação. No caso do visualizador tridimensional, este vai ser efectuado com uma comparação visual do modelo obtido pelo software AutoCAD da AutoDesk contra o visualizador integrado na prova de conceito desenvolvida. Em relação à tabela que faz a correspondência das cores presentes no desenho com um material, a comparação vai ser

efectuada novamente de forma visual, mas desta vez com uma listagem obtida no software atrás referido.

#### 6.2.2.1. VISUALIZAÇÕES – VISUALIZADOR TRIDIMENSIONAL

Este teste consiste em validar, qual o desempenho do visualizador tridimensional em representar um desenho definido em DWG. Os detalhes deste teste podem ser encontrados no ANEXO C, secção C.2.1 – Visualizações – Visualizador tridimensional (Página 135), sendo apresentado de seguida um resumo da sua execução.

TABELA 6.4 – DETALHE DO TESTE DE VISUALIZAÇÕES – VISUALIZADOR TRIDIMENSIONAL.

Teste	
<b>Nome do teste:</b>	Visualizador tridimensional
<b>Grupo:</b>	Personalização de vistas
<b>Propósito:</b>	Verificar o funcionamento do visualizador tridimensional
<b>Comentários:</b>	O ficheiro DWG utilizado vai ser da versão 2004
Comportamento	Veredicto
! Criar um desenho no AutoCAD	
! Gravar o desenho obtido no formato DWG versão 2004	
! Abrir o ficheiro obtido na implementação do sistema	
? Ficheiro aberto com sucesso	
? Desenho no visualizador correspondente ao desenhado	Sucesso
? Desenho no visualizador sem correspondência ao desenhado	Falha
? Ficheiro não abriu	Falha

Este teste foi efectuado de forma análoga ao “Teste de morfismo – Elementos gráficos simples”, sendo inclusive utilizado o mesmo ficheiro de testes.

Mas, após a tentativa de abrir o ficheiro, a mesma foi falhada, o que depois de uma análise inicial, se verificou que o problema se deveu a que o visualizador incorporado na implementação não suportar objectos do tipo círculo, como tal, impossibilitou a apresentação do desenho. Levando a que este teste tenha tido o resultado de falhado.

#### 6.2.2.2. VISUALIZAÇÕES – VISTA TABULAR

Este teste consiste em validar, qual o desempenho da vista tabular em representar os materiais presentes no ficheiro DWG, bem como quais as cores que lhes estão associadas. Os detalhes deste teste podem ser encontrados no ANEXO C, secção C.2.2 – Visualizações – Vista tabular (Página 135), sendo apresentado de seguida um resumo da sua execução.



TABELA 6.5 – DETALHE DO TESTE DE VISUALIZAÇÕES – VISTA TABULAR.

Teste	
<b>Nome do teste:</b>	Vista tabular
<b>Grupo:</b>	Personalização de vistas
<b>Propósito:</b>	Verificar o funcionamento da vista tabular
<b>Comentários:</b>	O ficheiro DWG utilizado vai ser da versão 2004
Comportamento	Veredicto
! Criar um desenho no AutoCAD com objectos de cores e materiais diferentes	
! Gravar o desenho obtido no formato DWG versão 2004	
! Abrir o ficheiro obtido na implementação do sistema	
? Ficheiro aberto com sucesso	
? Materiais e cores da tabela correspondem com as desenhadas	Sucesso
? Tabela com informação errada	Falha
? Ficheiro não abriu	Falha

Este teste foi também executado de uma forma análoga ao “Teste de morfismo – Elementos gráficos simples” tal como o anterior, mas desta feita com uma adição, a inclusão da definição de materiais associados a objectos.

Do resultado obtido do teste verifica-se que só quatro dos cinco materiais definidos são apresentados na vista tabular. Após uma breve análise concluiu-se que tal se deveu ao facto de existirem dois materiais que estão associados à mesma cor, e como a tabela está definida para não apresentar cores repetidas, apenas um dos materiais apareceu associado à cor amarela. Apesar de se poder concluir que este visualizador funciona em conformidade com o esperado, pode-se também concluir que a utilização desta vista pode levar à perda de informação.

### 6.2.3. TESTES DA CONTEXTUALIZAÇÃO DA INFORMAÇÃO

Este teste consiste em validar, como a contextualização da informação se comporta, neste caso específico o agrupamento dos objectos por cores. Os detalhes deste teste podem ser encontrados no ANEXO C, secção C.3 – Testes da contextualização da informação (Página 137), sendo apresentado de seguida um resumo da sua execução.

TABELA 6.6 – DETALHE DO TESTE DA CONTEXTUALIZAÇÃO DA INFORMAÇÃO.

Teste	
<b>Nome do teste:</b>	Agrupamento dos objectos por cor
<b>Grupo:</b>	Contextualização da informação
<b>Propósito:</b>	Verificar o funcionamento da contextualização da informação
<b>Comentários:</b>	O ficheiro DWG utilizado vai ser da versão 2004
Comportamento	Veredicto
! Criar um desenho no AutoCAD com objectos de cores e materiais diferentes	
! Gravar o desenho obtido no formato DWG versão 2004	
! Abrir o ficheiro obtido na implementação do sistema	
? Ficheiro aberto com sucesso	
! Exportar com e sem a aplicação de regras o ficheiro para X3D	
? Exportação efectuada com sucesso	
! Comparação de ambos com a definição da regra	
? Os resultados de ambos foram diferentes	
? Regra aplicada com sucesso	Sucesso
? Regra sem efeitos	Falha
? Ambos tiveram o mesmo resultado	Falha
? Exportação falhada	Falha
? Ficheiro não abriu	Falha

A análise da contextualização da informação passa, em primeiro lugar pela exportação de um ficheiro sem se recorrer a regras de contextualização da informação e depois pela exportação de um outro ficheiro, mas já com regras definidas (neste caso o agrupamento de objectos em grupos de cores), comparando-se posteriormente ambos os ficheiros X3D resultantes, verificando-se se esse agrupamento foi efectuado com sucesso.

Mais uma vez o ficheiro de testes a ser utilizado é um ficheiro previamente criado e utilizado, desta feita no “Visualizações – Vista Tabular”. Analisando os resultados obtidos deste teste, verifica-se que a aplicação da regra contextual definida foi um sucesso. Apenas de notar que, visto o nome do grupo ser definido pelo material do primeiro objecto do grupo (de forma análoga à vista tabular), que esse nome não se encontra totalmente correcto, pois existem diversos materiais dentro do grupo amarelo, o que não invalida o agrupamento com sucesso dos objectos por cores.

### 6.3. VEREDICTO ACERCA DA CONFORMIDADE DA SOLUÇÃO

A primeira conclusão que se pode retirar dos testes efectuados, é que a prova de conceito conseguiu abrir com sucesso os diversos ficheiros de testes utilizados, demonstrando que o importador desenvolvido desempenha a sua função.

Ao nível dos morfismos definidos, pode-se concluir que a sua aplicação é um sucesso, conseguindo-se obter no X3D uma representação equivalente dos modelos definidos em DWG. A única falha ocorrida resulta da falta de expressividade do X3D em relação à direcção para onde a profundidade das linhas ocorre. Sendo que este, não é uma falha ao nível da conversão, mas sim ao nível de limitação na descrição do X3D.

Em relação aos visualizadores, verificou-se que existem alguns problemas na sua utilização. No caso do visualizador tridimensional, este não suporta todo o tipo de objectos, nomeadamente objectos com curvas (e.g. círculo e arco neste caso), o que leva à impossibilidade de utilização do mesmo, caso os desenhos tenham esses tipos de objectos. Caso não tenha objectos destes tipos, foi confirmado pelo autor que o mesmo se apresenta de forma equivalente ao visualizador externo utilizado.

A vista tabular, funciona na sua totalidade, mas concluiu-se que a sua utilização pode levar à apresentação de informação incompleta, visto que por definição a tabela agrupa todos os objectos por cores e apresenta o material associado, o que dá problemas quando uma cor tem objectos de diferentes materiais.

Faltando então fazer uma passagem pelos testes da contextualização da informação, onde as regras para a contextualização da informação foram aplicadas com sucesso.

Pode-se então definir como veredicto final acerca da conformidade da solução face aos seus objectivos, que visto se estar perante uma prova de conceito, os testes foram um sucesso. Esta conclusão deve-se ao facto de os problemas encontrados, estarem ligados directamente com problemas técnicos de implementação ou de tecnologia, e não de nível conceptual, que era o nível onde se queria tirar uma conclusão.

---



## 7. CONCLUSÕES E PERSPECTIVAS FUTURAS

Com este trabalho pretendeu-se demonstrar a relevância do paradigma da visualização como um meio para melhorar significativamente a integração dos intervenientes, sejam empresas, indivíduos ou sistemas, ao longo de todo o ciclo de vida de um produto, aplicado num âmbito da colaboração em ambientes industriais. Nesta dissertação, a adopção do paradigma da visualização é proposto inserido num esforço global existente para a aplicação de modelos ao mundo industrial, denominado de empresas suportadas por modelos (MBE).

Este esforço de aplicação do paradigma da visualização no âmbito do MBE, permite que, num futuro, estando as empresas a descrever a sua informação de negócio de uma forma abstracta (recorrendo aos modelos), estas possam usufruir das vantagens da utilização do paradigma da visualização sem terem de proceder à alteração aos seus processos de negócio e às ferramentas que utilizam (PDES, Inc., 2007).

A utilização desta solução permite uma flexibilidade de utilização variada, como ser utilizada para quebrar as barreiras existentes na colaboração entre sistemas heterogéneos, agindo como um intermediário para converter a informação que vem de uma forma interpretável pelo sistema de origem numa forma que o sistema de destino possa interpretar e utilizar, sem a necessidade de intervenção humana a agir de mediador semântico neste processo.

Outra utilização advém da aplicação diferentes tipos de artefactos visuais à informação trocada, de forma a apresentá-la da forma mais apropriada ao tipo de informação existente (e.g. informação geográfica apresentada numa mapa). Recorrendo-se portanto à utilização de diversos tipos de artefactos visuais pode-se apresentar a informação adaptada ao domínio de aplicação de uma forma simples e de fácil interpretação.

Por fim, a última grande utilização que esta solução permite é o tratamento da informação. Esse tratamento pode ser utilizado de forma combinada com as anteriores, para vários fins como, a contextualização da informação, apresentando a informação estritamente necessária e aplicada ao domínio das funções dos utilizadores desta solução, ou como para filtrar a informação trocada, de forma a controlar a partilha, e.g. protegendo a propriedade intelectual da informação partilhada.

---

Estas três grandes formas de utilização desta solução contribuem para a realização do paradigma da visualização, sendo que a aplicação de artefactos visuais é um dos factores principais do paradigma da visualização, sendo que as outras agem como um meio para este grande objectivo (garantindo a interpretação e compreensão dos dados, bem como a sua contextualização).

A solução proposta neste trabalho tem o intuito de realizar o paradigma da visualização orientada pelos modelos, e com esta contribuir essencialmente para uma melhor interoperabilidade, promovendo uma melhor comunicação, contextualização e interpretação da informação, realizada segundo uma base formal, permitindo ainda uma facilidade na futura expansão de formatos e vistas.

A longo prazo, a aplicação generalizada deste paradigma, enquadrado no já referido BEM, necessita que exista um esforço em formalizar os formatos e vistas, para que se consiga automatizar todo este processo. O maior desafio neste campo trata-se de formalizar os formatos e vistas já existentes, pois é necessário um esforço extra para os formalizar já que nem sempre existe boa documentação sobre os mesmos.

Nos novos desenvolvimentos de formatos e vistas, o caminho a seguir será, de futuro, a utilização da abordagem do desenvolvimento orientado pelos modelos como um método de modelação e desenvolvimento, utilizado como uma evolução às actuais linguagens de programação, permitindo uma grande facilidade de aplicação de soluções como a proposta nesta dissertação. Utilizando portanto o MDD desde as fases iniciais de desenvolvimento dos formatos e vistas, ou mesmo de outros artefactos onde estas soluções se possam enquadrar, permite que, mais tarde, estas se apliquem sem grande esforço adicional.

Existem também novas formas de trabalho onde se podem tirar partido destas abordagens orientadas pelos modelos. Uma delas é os ambientes de trabalho (Workplaces) personalizados pelos intervenientes, pois estes querem ter acesso a toda a informação necessária para o desenrolar das suas funções num único ponto de acesso. Os modelos quando aplicados a este conceito lança um novo paradigma dos ambientes de trabalho gerados por modelos (em inglês, "Model-Generated Workplaces") (Rolfen, et al., 2007).

Estes ambientes de trabalho quando suportados em modelos podem ser facilmente adaptados às necessidades de cada um. A solução proposta nesta dissertação enquadra-se

---

nestes objectivos, contribuindo para a contextualização da informação proveniente dos vários sistemas que convergem num meio comum de utilização, permitindo que esta seja apresentada de forma adaptada às necessidades dos intervenientes.

Outro conceito, que está ligado com estes ambientes de trabalho é o “Software como um Serviço “ (do inglês, “Software as a Service – SaaS”), que tem como objectivo ter as ferramentas em servidores gerais (como as pesadas aplicações CAD), e estas serem disponibilizadas como serviços, sendo a solução proposta um facilitador para a integração deste conceito nos ambientes de trabalho personalizados pelos próprios intervenientes.

A arquitectura proposta neste dissertação, espera-se que seja utilizada como uma base para o futuro desenvolvimento de ferramentas, e não seja apenas só integrada nas ferramentas actuais, pois esta permite uma fácil conversão de vários modelos de dados, de aplicação de vistas sobre os mesmos, e de contextualizar os mesmos. Espera-se também que esta arquitectura possa contribuir não só para a aplicação a formatos e vistas, mas também ser expandida a outros campos aplicativos que contém com os mesmos tipos de problemas identificados nesta dissertação.

No fim desta implementação uma questão pode ser levantada, o facto de apenas se ter utilizado ferramentas automáticas para a passagem dos modelos para código na implementação da base comum. Isto deveu-se ao facto de as outras partes da implementação não estarem descritas em modelos, e como era necessário proceder à criação dos modelos de acordo com o código das restantes partes, passou-se por cima da passagem automática de modelos para código e trabalhou-se directamente no mesmo.

Portanto, um dos trabalhos mais relevantes a ser efectuados de futuro, passa pela aplicação de toda a abordagem do desenvolvimento orientado por modelos a todas as componentes da solução, abolindo por completo a necessidade da utilização da programação clássica (i.e. programar directamente numa linguagem de programação).

Do ponto de vista científico, e dada a contínua evolução dos sistemas industriais, i.e. novas versões dos formatos de dados surgem com diferentes modelos de informação, um dos grande desafios para o futuro é garantir a adaptabilidade dos sistemas a estas evoluções (i.e. as transformações adaptem-se às evoluções dos modelos), garantindo que de uma forma

---

automática, a evolução dos modelos existentes não ponha em causa o funcionamento da solução (Poole, 2001).

Existem mais desafios para o futuro a um nível científico, mas desta feita com uma grande vertente aplicacional, na utilização desta solução num modelo federado de interoperabilidade. A diferença da aproximação federada, para a unificada (base do desenvolvimento desta solução) encontra-se na definição da base comum, neste novo modelo a base comum não é imposta globalmente, logo, podem existir vários sistemas a interoperar com bases comuns distintas. Portanto, um estudo sobre esta aplicabilidade deveria ser efectuado, para preparar a utilização desta solução aos novos ambientes emergentes, que se enquadram nos modelos federados de interoperabilidade.

De frisar que o resultado deste trabalho foi utilizado no projecto de R&D co-financiado pela comissão europeia no contexto do sexto programa quadro, nº 31139, com o acrónimo INNOVAFUN, no âmbito na indústria do mobiliário. Aqui, a solução foi implementada numa ferramenta para a descrição de catálogos de produtos, designada de CADEF, provando a sua aplicabilidade em ambiente industrial. Também do envolvimento deste trabalho neste projecto saíram vários exemplos de modelos geométricos, que foram utilizados para um melhor desenvolvimento da solução trabalhada.

Este trabalho conta ainda com um seguimento aplicacional, (a aplicação de toda esta solução numa abordagem totalmente orientada pelos modelos) num projecto de indústria financiado pelo QREN – Quadro de Referência Estratégico Nacional, denominado BuildingStudio, promovido pela tecnológica CXS Computing S.A. ([http://www.porlisboa.gren.pt/np4/file/130/lista\\_beneficiarios.pdf](http://www.porlisboa.gren.pt/np4/file/130/lista_beneficiarios.pdf)).

---



## 8. BIBLIOGRAFIA

Adobe Systems Incorporated. (2008). *Adobe - Why PDF?* Obtido em 17 de Abril de 2008, de <http://www.adobe.com/products/acrobat/adobepdf.html>

Autodesk, Inc. (2008). *Autodesk - DWF Community - DWF: The Best File Format for Published Design Information.* Obtido em 17 de Abril de 2008, de <http://usa.autodesk.com/adsk/servlet/item?siteID=123112&id=8675679>

Autodesk, Incorporated. (20 de Março de 2007). *Unofficial AutoCAD History Pages.* Obtido em 17 de Abril de 2008, de [http://betaprograms.autodesk.com/history/autocad\\_release\\_history.htm](http://betaprograms.autodesk.com/history/autocad_release_history.htm)

Baker, M. J. (2008). *Maths - Angle between vectors.* Obtido em 21 de Outubro de 2008, de <http://www.euclideanspace.com/maths/algebra/vectors/angleBetween/index.htm>

Bentley Systems, Incorporated. (2008). *Bentley's OpenDGN Initiative.* Obtido em 17 de Abril de 2008, de <http://www.bentley.com/en-US/Products/MicroStation/OpenDGN/>

Bull, R. I. (2006). Integrating Dynamic Views Using Model Driven Development. *Proceedings of the 2006 conference of the Center for Advanced Studies on Collaborative research.* Toronto, Canada: ACM.

Bull, R. I., & Favre, J.-M. (2005). Visualization in the Context of Model Driven Engineering. *International Conference on Model Driven Engineering Languages and Systems.* Montego Bay, Jamaica: MODELS.

Bull, R. I., Storey, M.-A., Favre, J.-M., & Litoiu, M. (2006). An Architecture to Support Model Driven Software Visualization. *International Conference on Program Comprehension.* Atenas, Grécia.

CADAZZ. (2004). *CAD software history.* Obtido em 17 de Abril de 2008, de the best CAD software history on the Web: <http://www.cadazz.com/>

Dassault Systèmes. (15 de Junho de 2005). *Dassault Systemes Delivers 3D XML Specifications and Player.* Obtido de <http://www.3ds.com/news-events/press-room/release/899/1/>

---

Dassault Systemes. (s.d.). *Dassault Systemes: Format*. Obtido em 8 de Setembro de 2008, de <http://www.3ds.com/products/3dvia/3d-xml/00/>

Delgado, M. A. (2008). *Harmonisation of STEP and MDA conceptual models using Model Morphisms*. Dissertação de Mestrado, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, Departamento de Engenharia Informática.

ECMA. (Junho de 2007). Standard ECMA-363 4th Edition / June 2007 - Universal 3D File Format.

Faure, C., & Allilaire, F. (Julho de 2007). *The Tree to List example*. Obtido em 14 de Novembro de 2008, de ATL Basic Examples and Patterns: [http://www.eclipse.org/m2m/atl/basicExamples\\_Patterns/article.php?file=Tree2List/index.html](http://www.eclipse.org/m2m/atl/basicExamples_Patterns/article.php?file=Tree2List/index.html)

Gallaher, M. P., O'Connor, A. C., Dettbarn Jr., J. L., & Gilday, L. T. (2004). *Cost Analysis of Inadequate Interoperability in the U.S. Capital Facilities Industry*. NIST.

Hammond, J., Koubek, R. J., & Harvey, C. M. (2001). Distributed collaboration for engineering design: A review and reappraisal. *International Journal of Human Factors in Manufacturing*, 11 (1), 35-52.

IEEE. (1990). *IEEE Standard Computer Dictionary*. The Institute of Electrical and Electronics Engineers.

International Engineering Consortium. (2007). *WPF: Tree and Tabular Combined Notation (TTCN)*. Obtido em 5 de Outubro de 2008, de IEC: On-Line Education: <http://www.iec.org/online/tutorials/ttcn/>

ISO - International Organization for Standardization. (Maio de 2000). Product data representation and exchange: Integrated generic resource: Geometric and topological representation.

ISO 14258. (14 de Abril de 1999). Industrial Automation Systems—Concepts and Rules for Enterprise. ISO TC184/SC5/WG1.

---

Kai, C. C., Jacob, G. G., & Mei, T. (1997). Interface Between CAD and Rapid Prototyping Systems. Part 1: A Study of Existing Interfaces. *The International Journal of Advanced Manufacturing Technology*, 13, 566-570.

Khemlani, L. (15 de Fevereiro de 2006). Building the Future. AECbytes.

King, D. (2007). Product Lifecycle Management Matrix. Society of Manufacturing Engineers - Product Lifecycle Management Technical Group.

Langlois, B., & Exertie, D. (2005). MDSOFA: a Model-Driven Software Factory. *Best Practices for Model Driven Software Development*.

Luo, R. C., Lan, C. C., Tzou, J. H., & Lin, H. C. (2004). E-Manufacturing Web-Based Desktop Rapid Prototyping. *Proceedings of the 5th World Congress on Intelligent Control and Automation* (pp. 15-19). Hangzhou: IEEE.

MacKrell, J. (2006). *Extracting Value from Your Product and Process Data*. CIMdata, Inc.

MacKrell, J. (2006). *The Role of Product Data in Creating Innovation*. CIMdata.

Mankins, J. C. (6 de Abril de 1995). Technology Readiness Levels. Advanced Concepts Office - Office of Space Access and Technology - NASA.

NIST. (Maio de 2002). *The Initial Graphics Exchange Specification (IGES)*. Obtido em 13 de Setembro de 2008, de <http://ts.nist.gov/standards/iges/>

Onofre, S. M. (2007). *Plataforma para testes de conformidade de sistemas baseados em módulos conceptuais STEP*. Dissertação de Mestrado, Universidade Nova de Lisboa - Faculdade de Ciências e Tecnologia, Departamento de Engenharia Electrotécnica.

Open Design Alliance. (2008). *OpenDWG History | Open Design Alliance*. Obtido em 17 de Abril de 2008, de <http://www.opendesign.com/node/86>

Open Design Alliance. (2008). *OpenDWG R13/R14/R2000/R2004/R2007 File Format Specification Version 4.0*.

PDES, Inc. (5 de Dezembro de 2007). The Model Based Enterprise (MBE). *Defense Manufacturing Conference*. Las Vegas, Nevada, Estados Unidos da America.

---

Poole, J. D. (2001). Model-Driven Architecture: Vision, Standards And Emerging Technologies. *ECOOP 2001*.

Rolfesen, R. K., Boell, D., Pronios, C., Knothe, T., Anastasiou, M., Elvesæter, B., et al. (2007). Model-Generated Workplaces: An Interoperability Approach. *Enterprise Interoperability II*, (pp. 771-782).

Rothman, A. (29 de Setembro de 2006). Airbus Vows Computers Will Speak Same Language After A380 Delay. Bloomberg.

Schafersman, S. D. (Janeiro de 1994). *Scientific Thinking and the Scientific Method*. Obtido em 10 de Setembro de 2008, de An Introduction to Science: <http://www.freeinquiry.com/intro-to-sci.html>

Schilli, B., & Dai, F. (2006). Collaborative life cycle management between suppliers and OEM. *Comput. Ind.*, 57 (8), 725-731.

Schmidt, D. C. (Fevereiro de 2006). Model-Driven Engineering.

Siemens Product Lifecycle Management Software Inc. (2008). *Parasolid: Siemens PLM Software*. Obtido em 8 de Setembro de 2008, de [http://www.plm.automation.siemens.com/en\\_us/products/open/parasolid/index.shtml](http://www.plm.automation.siemens.com/en_us/products/open/parasolid/index.shtml)

Spatial Corp. (2007). *Spatial:3D ACIS Modeler*. Obtido em 17 de Abril de 2008, de <http://www.spatial.com/products/acis.html>

Sun Microsystems, Inc. (s.d.). *About the Java Technology*. Obtido em 14 de Setembro de 2008, de The Java Tutorials: <http://java.sun.com/docs/books/tutorial/getStarted/intro/definition.html>

Sun Microsystems, Inc. (2008). *How to Use Lists*. Obtido em 14 de Novembro de 2008, de The Java Tutorials: <http://java.sun.com/docs/books/tutorial/uiswing/components/list.html>

Sun Microsystems, Inc. (2008). *How to Use Trees*. Obtido em 14 de Novembro de 2008, de The Java Tutorials: <http://java.sun.com/docs/books/tutorial/uiswing/components/tree.html>

Tretmans, J. (25 de Janeiro de 2001). An Overview of OSI Conformance Testing. University of Twente.

---

---

UGS Corp. (2008). *JT File Format Reference Version 8.1 Rev-B*.

UGS Corporation. (2008). *Parasolid XT Format Reference*.

US PRO. (1996). Initial Graphics Exchange Specification 5.3. U.S. Product Data Association.

Web3D Consortium. (s.d.). *What is X3D?* Obtido em 17 de Abril de 2008, de Web3D Consortium - Royalty Free, Open Standards for Real-Time 3D Communication: <http://www.web3d.org/about/overview/>

Web3D Consortium, Inc. (Dezembro de 2007). Information technology — Computer graphics and image processing — Extensible 3D (X3D). *ISO/IEC FDIS 19775-1:2008* .

Wijk, J. J. (2006). Views on Visualization. *Visualization and Computer Graphics, IEEE Transactions on* , 12 (4), 1000 - 433.

Willaert, S., Graaf, R. d., & Minderhoud, S. (1998). Collaborative engineering: A case study of Concurrent Engineering in a wider context. *Journal of Engineering and Technology Management* , 87-109.

Wu, T., & Cheung, E. H. (2006). Enhanced STL. *The International Journal of Advanced Manufacturing Technology* , 29, 1143-1150.

---



## ANEXO A. NORMAS PARA A REPRESENTAÇÃO GEOMÉTRICA

Para a representação geométrica de modelos 3D existem vários formatos de transferência de dados com capacidades distintas. Segue-se uma apresentação sucinta de um conjunto desses formatos, considerados os mais relevantes, quer seja pela capacidade de descrição dos modelos 3D ou pela aceitação por parte da indústria.

- **3D XML**

Formato proprietário de uma das empresas de renome na indústria de desenvolvimento de software de desenvolvimento de produtos, existe desde Outubro de 2004 e foi desenvolvido com base em seis pontos-chave (Dassault Systemes):

- Utilização: com o uso deste formato a informação 3D pode ser facilmente incorporada em documentação técnica, manuais de manutenção, brochuras de marketing, websites, trocas de correio electrónico e mais um sem número de tarefas do dia a dia. Esta reutilização da informação 3D alarga o número de utilizadores com acesso a modelos 3D.
  - Abertura: o conteúdo do formato está totalmente descrito utilizando um esquema de XML, permitindo aos utilizadores implementar o formato nas suas próprias aplicações e também estende-lo/personaliza-lo para ir ao encontro das suas necessidades.
  - Fácil de adoptar: devido à adopção do XML globalmente, este formato pode ser facilmente lido ou escrito pelas inúmeras aplicações já existentes facilitando a interpretação dos dados. A adopção deste formato reduz os custos da reutilização de dados 3D.
  - Colaboração: este novo formato melhora a colaboração em torno da informação 3D. Com ele os dados podem ser integrados com o resto dos processos de negócio da companhia. Informação do desenvolvimento do produto criada nos departamentos de engenharia está agora acessível por toda a empresa.
  - Leve e partilhável: os utilizadores podem capturar e partilhar dados 3D de uma forma rápida e fácil sobre um formato leve, com tamanhos até 90 por cento mais pequenos que os formatos existentes, permitindo serem carregados e transferidos através da internet e usados no contexto dos sistemas colaborativos.
-

- Totalmente documentado: a documentação pública consiste num conjunto de esquemas orientados para as necessidades de aplicação e num manual de utilizador para um fácil entendimento da informação dos esquemas.

- **ACIS – SAB (Standard ACIS Binary) e SAT (Standard ACIS Text)**

Existente desde 1989, está-se perante algo mais que apenas um formato de troca de modelos, mas sim um “kernel” de modelação 3D, este permite aos fabricantes trabalharem directamente nas funcionalidades de modelação 3D, tais como a modelação por redes, superfícies e sólidos. Essas funcionalidades são apresentadas através de uma estrutura livre, baseada na arquitectura orientada a objectos do C++. A partir do ano 2000, após a empresa dona do formato (Spatial Corp.) ter sido adquirida por a “Dassault Systèmes”, o formato foi alterado, e nunca mais foi distribuído de forma pública (CADAZZ, 2004).

Neste documento, visto ele recair sobre os formatos de troca de modelos 3D, vai-se dar importância a essa parte do “kernel”. Existem duas formas de representar o ficheiro de dados, a forma de texto e a binária. Estes são o formato SAT e SAB respectivamente, sendo a informação trocada neles a mesmas em ambos os casos.

O tipo de informação trocada nos ficheiros, depende directamente da forma com que o ACIS modela os objectos, permitindo por exemplo, a conjugação de várias curvas 2D de forma a criar superfícies ou sólidos complexos, gerar padrões de formas repetidas, definir grossura de superfícies, fazer operações geométricas sobre os objectos definidos, intersectar, subtrair, unir qualquer combinação de curvas, superfícies ou sólidos, etc.

Outra funcionalidade interessante do formato de ficheiros do ACIS, é o suporte do histórico de operações, que permite trocar a forma como o modelo foi criado pelo seu desenhador, permitindo que se navegue nos passos de criação de forma a alterar o processo de desenho. Por exemplo, imaginemos um modelo que conta com a personalização de uma marca, caso exista o histórico de como essa personalização foi inserida, facilmente se consegue obter o modelo inicial de forma a se proceder à nova personalização.

Existem várias aplicações de CAD, que contam com a modelação ACIS, para descrever os seus objectos, como por exemplo, um dos mais conhecidos do mercado, o AutoCAD® (Spatial Corp., 2007).

---



- **DGN (Design)**

Formato utilizado pelo “Bentley Systems MicroStation” é em tudo semelhante ao DWG da Autodesk, descrito mais à frente. Mas apesar de não ser amplamente utilizado como o DWG tem a grande vantagem de ser documentado publicamente.

O nível de elementos suportado é o das formas de modelação 3D mais utilizadas. Este formato tem características interessantes, como o facto de suportar vários modelos em simultâneo, referências entre entidades, escalas, etc.

Este é guardado numa forma binária, o que impossibilita a sua leitura sem se recorrer a um leitor apropriado (Bentley Systems, Incorporated, 2008).

- **DWF (Design Web Format)**

Este formato, desenvolvido pela Autodesk em 1995 como parte do projecto “WHIP!” (Um plug-in para o Netscape Navigator), originalmente conhecido como “Drawing Web Format” e só utilizado no AutoCAD®, passou, por culpa da sua aceitação, para outras plataformas e passou a contar com o nome actual. Também com o passar do tempo foi-lhe incorporado o suporte de modelos 3D.

O DWF é um formato aberto, desenvolvido com o objectivo de uma distribuição eficiente e segura das visualizações, para quem necessite de ver, de rever, ou de imprimir os projectos. Estes ficheiros têm uma elevada taxa de compressão, pois dispensa, uma grande parte da informação complexa associada aos modelos CAD. Outra característica é o facto de o conteúdo original ser protegido, fazendo com que as representações trocadas sejam alteradas (Autodesk, Inc., 2008).

- **DWG (Drawing) e DXF (Drawing Exchange Format)**

Originário do final da década de 70 como o formato nativo da primeira aplicação CAD baseada em microcomputadores foi, no início da década de 80, tornado a base do software CAD mais vendido da história, o AutoCAD®.

Apesar de este ser o formato mais utilizado na indústria, é um formato proprietário, e a sua criadora, a Autodesk, desde o início que mantém privada a sua especificação. Para colmatar este problema, foi criado o formato DXF, que conta com a especificação pública, sendo a sua

---

estrutura igual à do DWG, diferindo no facto de este formato ser uma representação ASCII do mesmo. Mas outro problema surgiu, o AutoCAD® continuou a evoluir e certos tipos de objectos, como os sólidos ACIS, as regiões e os blocos dinâmicos, deixaram de ser documentados, portanto o DXF perdeu a sua equivalência de uma forma ASCII em relação ao DWG.

Visto se estar perante o formato mais utilizado na indústria, era importante este poder ser utilizado não só pelo AutoCAD®, mas pelas mais variadas ferramentas CAD existentes. Para tal, em 1998, foi criada a “Open Design Alliance”, com o objectivo de distribuir de forma livre métodos para a utilização do formato DWG (Open Design Alliance, 2008).

O DWG não é apenas uma simples representação de objectos, ele é uma pequena base de dados de modelos, contendo informação de camadas, materiais associados, relação entre objectos, blocos e entidades. Este formato suporta a grande parte dos formalismos de modelação 3D, sendo que a representação de geometria quase não tem limites, devido à utilização do “kernel” de modelação ACIS para a representação de sólidos (Autodesk, Incorporated, 2007).

- **IGES (Initial Graphics Exchange Standard)**

Um formato criado de forma a se uniformizar a troca de dados entre os vários Sistemas CAD existentes, que conta com a sua primeira publicação da norma apresentada em Janeiro de 1980. Este formato é um dos mais utilizados hoje em dia, devido ao facto que desde 1988 o Departamento de Defesa dos Estados Unidos obriga a que toda a informação relativa a sistemas de armamento esteja em formato electrónico, especificamente no formato IGES.

Hoje em dia existem plug-in para browsers de internet que permitem visualizar ficheiros criados à vinte anos atrás. O interesse da continuação do desenvolvimento deste formato acabou, aquando o primeiro lançamento do STEP<sup>10</sup>, em 1994, tendo sido o ultimo lançamento do IGES, a versão 5.3 do ano de 1996. Após uma década o STEP ainda não conseguiu cumprir a sua promessa de substituir o IGES, continuando este a ser o formato mais utilizado na troca de informação CAD.

---

<sup>10</sup> STEP - Standard for the Exchange of Product model data – Nome oficial da Norma ISO 10303.

---

De início, só eram suportadas modelações bidimensionais, actualmente é possível também representar superfícies e sólidos, o que alarga o formato ao mundo tridimensional. Devido ao facto de este formato ser bastante generalista, e suportar um conjunto bastante alargado de características para suportar os conceitos existentes nos sistemas CAD, cada sistema fez a sua própria interpretação da norma, tendo-se perdido a sua maior potencialidade, que seria a interoperabilidade entre sistemas, passando a existir a necessidade de em cada sistema, aquando a exportação do modelo, ser necessário o sistema CAD de destino. A dimensão dos ficheiros também é um ponto contra, apesar de os ficheiros serem descritos em formato ASCII, contêm uma estrutura bastante complexa, e são quase impossíveis de serem lidos directamente por um humano, tal como se pode verificar na tradução do desenho da Figura A.1 no ficheiro resultante na Figura A.2.

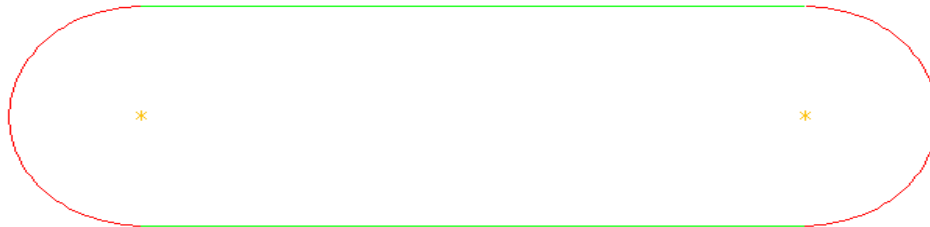


FIGURA A.1 – DESENHO A REPRESENTAR.

```

1H, ,1H; ,4H$LOT,37H$1$DUA2:[IGESLIB.BDRAFT.B2I]SLOT.IGS; ,          S          1
17HBravo3 BravODRAFT,31HBravo3->IGES v3.002 (02-Oct-87),32,38,6,38,15,   G          1
4H$LOT,1, ,1,4HINCH,8,0.08,13H871006.192927,1.E-06,6, ,           G          2
31HD. A. Harrod, Tel. 313/995-6333,24HAPPLICON - Ann Arbor, MI,4,0;    G          3
    116      1      0      1      0      0      0      0      0      1D          4
    116      1      5      1      0      0      0      0      0      OD          1
    116      2      0      1      0      0      0      0      0      1D          2
    116      1      5      1      0      0      0      0      0      OD          3
    100      3      0      1      0      0      0      0      0      1D          4
    100      1      2      1      0      0      0      0      0      1D          5
    100      4      0      1      0      0      0      0      0      OD          6
    100      1      2      1      0      0      0      0      0      1D          7
    110      5      0      1      0      0      0      0      0      OD          8
    110      1      3      1      0      0      0      0      0      1D          9
    110      6      0      1      0      0      0      0      0      OD          10
    110      1      3      1      0      0      0      0      0      1D          11
    110      1      3      1      0      0      0      0      0      OD          12
116,0.,0.,0.,0,0,0,0;          1P          1
116,5.,0.,0.,0,0,0,0;          3P          2
100,0.,0.,0.,0.,1.,0.,-1.,0,0; 5P          3
100,0.,5.,0.,5.,-1.,5.,1.,0,0; 7P          4
110,0.,-1.,0.,5.,-1.,0.,0,0;   9P          5
110,0.,1.,0.,5.,1.,0.,0,0;     11P         6
S      1G      4D      12P      6          T          1

```

FIGURA A.2 – FICHEIRO IGES RESULTANTE.

Como se pode verificar a complexidade derivada de um ficheiro IGES, é bastante alta, o que leva a impossibilidade da sua interpretação por um humano sem recorrer a ferramentas que permitam uma fácil visualização do mesmo (US PRO, 1996).

- **JT**

Este formato, orientado para a indústria, de alto desempenho, leve e flexível tem como objectivo a captura e redistribuição de dados 3D dos produtos de forma a permitir a colaboração, validação e visualização em toda a empresa. Sendo que o mesmo é um standard *de facto* para a visualização 3D na indústria automóvel, e um dos formatos mais dominantes nos outros domínios.

Sendo tanto robusto, como *streamable*, e contando com uma elevada taxa de compressão para uma representação compacta e eficiente, este formato foi desenhado para ser facilmente incorporado nas soluções das empresas, produzindo um único conjunto de recursos 3D que suportam toda a gama de processos de download de informação, indo desde uma visão leve baseada em browsers Web até ao completo detalhe do produto.

Na realidade este formato é uma descrição de cenas com suporte para nós de CAD bem como para atributos. A informação de superfícies é guardada com o recurso a sofisticadas técnicas de compressão de geometria. Atributos visuais como luzes, texturas, materiais e sombras são suportados. Informação associada ao produto e à sua manufactura, definição de partes e meta-informação, tais como uma variedade de formas de representação são suportados. Também são suportados varias formas de distribuição incluído métodos assíncronos de *streaming* do conteúdo.

- **Parasolid XT**

A Parasolid é líder mundial na produção de componentes de software para a modelação 3D, fornecendo um “kernel” que permita que os utilizadores modelem de forma rápida e robusta os produtos mais complexos da indústria, tal como o seu concorrente, anteriormente descrito, o ACIS.

Com uma globalização das aplicações que utilizam o Parasolid como “kernel” de modelação, que conta com mais de dois milhões de utilizadores, criou-se a necessidade de um formato que permita a troca dos modelos criados sem a perda de informação, nascendo o formato nativo XT (Siemens Product Lifecycle Management Software Inc., 2008).

A informação transmitida, tanto pode ser geométrica, definindo a forma dos objectos, como topológica, como as cotas, alturas, volumes, etc. O formato conta com uma especificação

---

publica que pode ser acedida gratuitamente, e que permite a interpretação dos dados transmitidos sem se ter de recorrer ao “kernel” de modelação (UGS Corporation, 2008).

- **PDF (Portable Document Format)**

Introduzido em 1993, é neste momento o formato de troca de documentos mais utilizado na internet. O conteúdo destes ficheiros passa por texto, gráficos vectoriais e imagens.

Desde Novembro de 2004, que o formato suporta a inclusão de representações gráficas 3D, utilizando para a sua representação o U3D<sup>11</sup> (descrito abaixo). Estão disponibilizados “Toolkits” pela Adobe que permitem a fácil criação de ficheiros através de qualquer aplicação já existente.

Um dos pontos fortes da distribuição de modelos 3D em PDF advém do facto de se poderem trocar modelos interactivos. Esta funcionalidade é bastante útil, por exemplo, no caso de o documento a enviar ser um manual de formação para o uso de uma máquina, onde se podem incluir animações para um mais fácil entendimento das instruções (Adobe Systems Incorporated, 2008).

- **SLC (Stereo Lithography Contour)**

O formato SLC, criado em 1994, pela empresa Americana 3D Systems, descreve objectos através da representação 2½D de cada uma das suas camadas, ou seja, o objecto é seccionado em várias camadas divididas em intervalos constantes. Cada camada contém a informação de quais os limites internos e externos do objecto. Outra característica relevante deste formato é o facto de ser caracterizado por uma representação ASCII.

Este formato tem grande utilidade aquando a sua utilização para a representação de dados obtidos por exemplo, através de TACs<sup>12</sup>, ressonâncias magnéticas, ou seja, imagiologia que trabalhe efectuando a leitura dos objectos através do seccionamento por camadas, como pode ser observado na figura seguinte (Kai, Jacob, & Mei, 1997).

---

<sup>11</sup> U3D – Universal 3D

<sup>12</sup> TAC – Tomografia Axial Computorizada – é um método de diagnóstico médico por imagem, que consiste numa imagem que representa uma secção ou “fatia” do corpo.

---

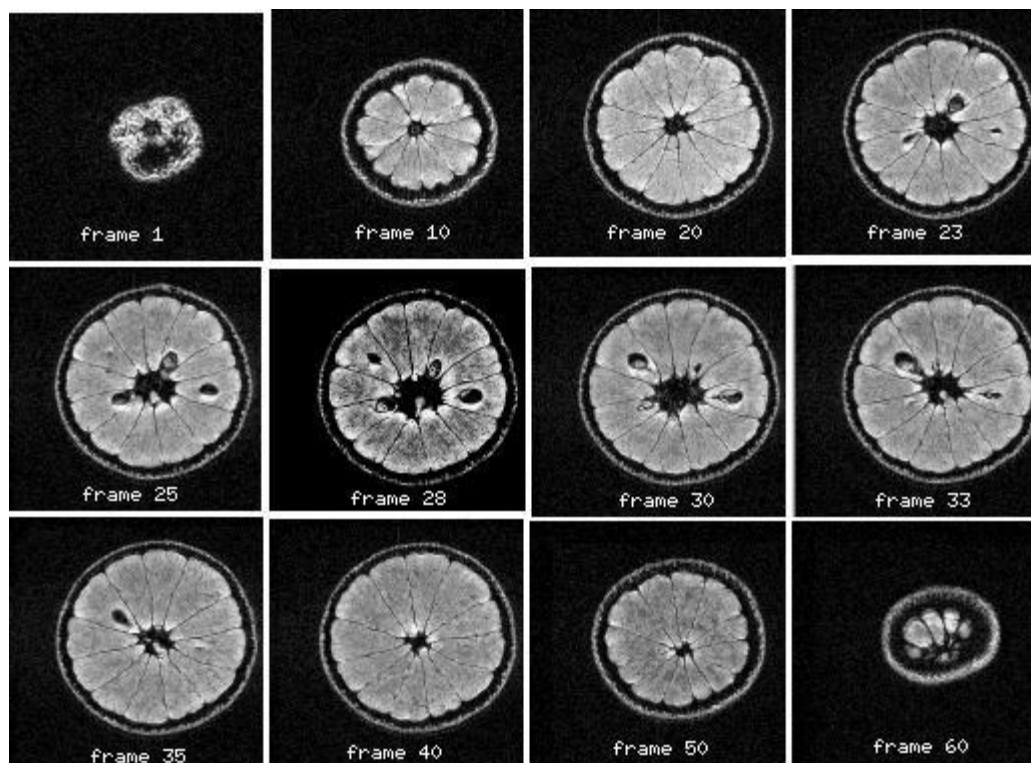


FIGURA A.3 – RESULTADO DE UMA RESSONÂNCIA MAGNÉTICA A UMA LARANJA.

- **STEP (STandard for the Exchange of Product model data)**

A norma STEP apresenta-se como um esforço de normalização de interfaces de troca de dados, ostentando o selo da ISO<sup>13</sup>. A sua principal característica, é não se estar perante apenas um formato mas sim toda uma norma que permite não só o suporte à representação das características geométricas de um objecto, mas também de todas as características de um produto durante o seu ciclo de vida.

O modelo de troca de dados é tão genérico que é possível inclusive trocar informação relativa a recursos e processos, aliada aos dados do produto. Este carácter geral permite abranger um sem número de indústrias, desde o automóvel até à aeronáutica. Para cada indústria em específico, existe um modelo definido para a modelação de produtos, representado por um AP (Application Protocol – Protocolo de Aplicação) que permite a sua descrição, aliando a representação geométrica às características específicas do sector de aplicação em questão.

---

<sup>13</sup> ISO - International Organization for Standardization – <http://www.iso.org/>

---

Cada AP encontra-se definido com base no formalismo expresso pela linguagem EXPRESS. Esta apresenta-se como uma linguagem de descrição, capaz de modelar entidades e relações numa metodologia orientada pelos objectos. A troca de informação é feita portanto num formato bem definido, designado por parte 21, onde se encontram descritos não só valores, mas também as relações de cada valor com a entidade correspondente no modelo de dados, por exemplo, se diz respeito a uma superfície, se é um ponto, se é uma recta, etc., existindo uma parte da norma onde é descrito o formalismo para a representação geométrica, sendo esta a parte 42 (ISO - International Organization for Standardization, 2000).

- **STL (Structured Triangular Language)**

Este formato, originalmente denominado por Stereolithography Tessellation Language, devido às suas características de, um formato aberto, de fácil utilização e com uma descrição simplista das formas, tornou-se uma norma na indústria, sendo actualmente o formato padrão na prototipagem rápida. Hoje em dia o STL é conhecido por Structured Triangular Language, nome que advém do facto de este formato representar os objectos descrevendo a sua superfície com base em em modelos triangulares.

Este formato tem duas formas de ser apresentado, a forma ASCII<sup>14</sup> e a binária, ambas representam a mesma informação, com a diferença da forma como a informação é guardada. Na forma ASCII este é um ficheiro de texto que pode ser alterado com um comum editor de texto, na forma binária, este é guardado num formato em código de máquina, que conta com a vantagem de ser de menor dimensão e de se obter um mais rápido processamento dos dados.

Na figura seguinte, pode-se observar como os triângulos definem um objecto.



FIGURA A.4 – EXEMPLO DA DECOMPOSIÇÃO DE UM OBJECTO EM TRIÂNGULOS.

---

<sup>14</sup> ASCII – American Standard Code for Information Interchange – Formato de codificação baseado no alfabeto inglês.

---

Na Figura A.5 é apresentado um fragmento de uma representação ASCII neste formato:

```

solid
  facet normal -9.82193649E-01 -1.87871352E-01 -0.00000000E+00
    outer loop
      vertex 2.74852419E+00 2.02480936E+00 0.04999995E+00
      vertex 2.74479361E+00 2.04428148E+00 0.05000019E+00
      vertex 2.74852419E+00 2.02480936E+00 2.55000019E+00
    endloop
  endfacet
  facet normal -9.99949455E-01 -1.00541785E-02 0.00000000E+00
    outer loop
      vertex 2.74852419E+00 2.02480936E+00 2.55000019E+00
      vertex 2.74877405E+00 1.99995899E+00 0.04999995E+00
      vertex 2.74852419E+00 2.02480936E+00 0.04999995E+00
    endloop
  endfacet
  ...
  facet normal 1.00000000E+00 0.00000000E+00 -0.00000000E+00
    outer loop
      vertex 4.49976158E+00 2.37496901E+00 1.54999995E+00
      vertex 4.49976158E+00 2.37496901E+00 2.30000019E+00
      vertex 4.49976158E+00 1.62496901E+00 2.30000019E+00
    endloop
  endfacet
endsolid

```

FIGURA A.5 – FRAGMENTO DE UM FICHEIRO STL, NO MODO ASCII.

Neste fragmento do ficheiro, de notar a representação dos, já referidos, triângulos, sendo também definida a normal à face para se definir qual a parte interna e externa do objecto (Wu & Cheung, 2006).

- **U3D (Universal 3D)**

Formato também criado para a tentativa de existência de um novo standard para a troca de representações 3D, muito certamente terá um futuro mais promissor, como a forma de representação utilizada pela Adobe no seu formato PDF.

Este consiste num formato extensível desenvolvimento para o *downstream* de modelos CAD e de visualização, para uma integração nas ferramentas de mais relevo na indústria. Algumas das características que mais se salientam são: a definição de uma arquitectura para a execução, que optimiza a modificação da geometria em tempo de execução; nível de detalhe contínuo; compressão específica para o domínio de aplicação; *streaming* de dados progressivos e animações (ECMA, 2007).

- **X3D (Extensible 3D)**

Norma ISO para a representação e troca de cenas 3D e objectos utilizando o XML<sup>15</sup>. Sucessor do VRML<sup>16</sup>, este formato fornece um sistema para a interacção de gráficos em tempo real.

<sup>15</sup> XML - Extensible Markup Language

<sup>16</sup> VRML - Virtual Reality Modeling Language



No âmbito deste documento interessa mais, as componentes da norma orientadas para o uso na engenharia, como o CAD.

Este formato, ao ter como base o XML, conta com uma grande facilidade de integração, como por exemplo, em “web services”, e pode ser lido, alterado ou escrito apenas recorrendo a um simples editor de texto. Tem também como característica o facto de ser extensível, o que permite que se adicionem novas funcionalidades de forma a ir ao encontro de todas as especificações que o exigente mercado necessita.

Os objectos 3D podem ser descritos com geometria poligonal, paramétrica e é possível aplicar-lhes transformações, definir luzes, especificar quais os materiais, etc. Existe também uma parte para a descrição de informação CAD, disponível desde a primeira revisão, que suporta camadas, faces, bem como grupos de objectos que compõem um modelo. Outras possibilidades interessantes são o facto de se poder definir a interacção com o utilizador, navegação na cena 3D, etc.

Tendo as potencialidades do XML por detrás, é possível estender as potencialidades desta tecnologia com, por exemplo, a aplicação de regras. Um caso onde estas se aplicam é na aplicação de testes de conformidade para, se poder, por exemplo, na utilização dos modelos em máquinas, verificar se as mesmas têm capacidades técnicas para os interpretar.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.0//EN" "http://www.web3d.org/specifications/x3d-3.0.dtd">

<X3D version='3.0' profile='Immersive'>

  <Scene>
    <!-- This NavigationInfo node is added to many scenes, making examination of objects easier. -->
    <Shape>
      <!-- When attribute values are not specified, the default values from the VRML Specification are used. -->
      <Appearance>
        <Material diffuseColor='.2 .8 .4'/>
      </Appearance>
      <Box size='1 3 5'/>
    </Shape>
  </Scene>
</X3D>
```

FIGURA A.6 – EXEMPLO DE UM FICHEIRO DE FORMATO X3D.

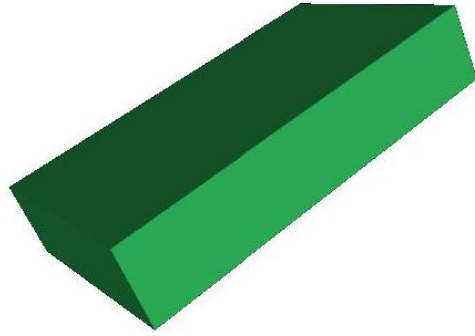


FIGURA A.7 – REPRESENTAÇÃO 3D DO OBJECTO DEMONSTRADO NA FIGURA A.6.

Outro factor a ter em conta é, ao ser um formato aberto e baseado em XML, facilmente um ser humano consegue verificar o seu conteúdo sem ter recorrer a aplicações externas, que nem estão disponíveis. Na Figura A.6 pode-se observar um exemplo da simplicidade de utilização deste formato, existindo ainda vários “Toolkit’s” livres para a visualização, criação, edição e interacção com os modelos. (Web3D Consortium).

---

## ANEXO B. MORFISMOS ENTRE O DWG E A BASE COMUM EM X3D

Para se definirem os mapeamentos entre ambos os formatos é necessário ter disponível a especificação de ambos os formatos, no caso do DWG a disponibilizada em (Open Design Alliance, 2008), e no caso do X3D a disponibilizada oficialmente em (Web3D Consortium, Inc., 2007). Estes mapeamentos vão ser definidos em duas fases, a primeira retrata os tipos básicos (tipos utilizados na descrição das entidades geométricas, e.g. pontos tridimensionais) e a segunda os compostos (e.g. objectos geométricos, como circulo).

Os mapeamentos não são apresentados em nenhuma linguagem de transformação, visto não estarem ambos definidos formalmente, o que leva a que todos estes mapeamentos se decorram apenas ao nível das instâncias sem terem em conta as diferenças existentes ao nível dos tipos. Também é de referir o facto de não serem apresentados todos os artefactos que são exportados na prova de conceito para X3D (e.g. texturas, e materiais), isto advém de este mapeamento não ser efectuado directamente do DWG mas sim do visualizador tabular.

### B.1. TIPOS BÁSICOS

Existem diversos tipos que são utilizados na descrição das várias entidades geométricas que o DWG contem, que irão ter a sua correspondência no X3D com ou sem processamento da informação, i.e. correspondência directa sem perda de informação, correspondência com processamento de informação com e sem perda de informação.

De seguida, vão ser apresentados os mapeamentos definidos. No caso dos tipos básicos definidos no X3D, estes têm um prefixo que define o número de valores desse tipo que são instanciados, i.e. se o tipo começar por SF (Single-valued Field, campo com um valor) apenas um valor desse tipo é definido, caso comece por MF (Multiple-valued Field, campo com múltiplos valores), são definidos vários valores do tipo escolhido (e.g. age como um vector).

De referir também que todas as dimensões do X3D são descritas no sistema métrico, e os ângulos em radianos. No caso do DWG este permite definir quais as unidades a se utilizarem, mas como esta vertente não foi implementada na prova de conceito, vai-se assumir que todas as unidades estão ao mesmo nível das do X3D, facilitando as conversões.

---

- **BitDouble → SFDouble**

Esta conversão é efectuada directamente, visto que em ambos os casos é utilizado o formato de vírgula flutuante de precisão dupla, para definir um único valor.

- **BitDouble → SFFloat**

Como por vezes, nos objectos equivalentes do X3D os mesmos dados que no DWG são representados em vírgula flutuante de precisão dupla, são representados apenas com precisão simples, é necessário contemplar essa conversão.

Visto serem ambos descritos no mesmo tipo de formato, vírgula flutuante, esta conversão é directa, contando apenas com uma limitação, baixando-se a precisão pode-se vir a estar perante a situação em que a precisão simples não é suficiente para representar o numero descrito pelo DWG, perdendo-se informação na conversão.

- **3D Point → SFVec3d**

No DWG um ponto tridimensional em coordenadas mundo e é definido como o conjunto de três BitDouble, portanto, tal como anteriormente, a conversão é directa, mas desta feita, para um outro tipo, o SFVec3d que define um vector tridimensional constituído por três valores do tipo SFDouble, nos mesmos padrões que o caso anterior.

- **3DPoint → SFVec3f**

Esta transformação está no mesmo âmbito da BitDouble para SFFloat pois, serve também para ser utilizada mais à frente na descrição dos objectos geométricos que utilizam este tipo para a descrição dos pontos, podendo-se perder informação na conversão tal como explicado anteriormente.

- **BitThickness → SFDouble**

Mais uma vez, esta conversão é análoga à primeira, contando com uma diferença, apenas na forma como o valor é descrito no DWG. De forma a comprimir o ficheiro é utilizado um prefixo, caso esse prefixo seja um bit de valor um, é assumido o valor 0.0 caso contrario este valor é descrito como um BitDouble normal.

---

- **BitThickness → SFFloat**

Tal como anteriormente, mais à frente na aplicação deste tipo nos objectos do X3D vai ser necessário a utilização dos valores em virgula flutuante de precisão simples, logo, esta conversão é análoga à anterior mas pode ocorrer a perda de informação.

- **BitExtrusion → SFRotation**

Equivalente ao ponto tridimensional, ao nível da representação de dados, conta apenas com a diferença de na descrição do DWG ser definido um valor por defeito, caso o valor seja iniciado por um bit de valor unitário, de 0, 0, 1. Nos objectos descritos mais à frente, este tipo de dados é utilizado para definir o sentido de orientação dos objectos, mas desta feita, a conversão para o tipo que define as orientações no X3D, o SFRotation, não é trivial.

O funcionamento do SFRotation, baseia-se, não num vector que aponta para o sentido de orientação do objecto, como no DWG, mas sim por um vector que define onde a rotação é efectuada, e um ângulo que define a quantidade de rotação aplicada. Estes são definidos na forma de quadro números no formato de vírgula flutuante  $(x, y, z, a)$ , onde os quatro primeiros definem o vector de rotação e o  $a$ , o valor de rotação pela regra da mão direita em torno do eixo, descrito em radianos.

A matriz 3x3 que representa a rotação  $(x, y, z, a)$  é a seguinte:

$$\begin{bmatrix} tx^2 + c & txy + sz & txz - sy \\ txy - sz & ty^2 + c & tyz + sx \\ txz + sy & tyz - sx & tz^2 + c \end{bmatrix}$$

$$\text{onde } c = \cos a, s = \sin a \text{ e } t = 1 - c$$

É portanto necessário encontrar o eixo de rotação e o ângulo necessário de aplicar, aos objectos criados para estes resultarem os originais. Para tal são necessários dois vectores, o inicial, para onde o objecto está a apontar, no caso do X3D é o vector 0,0,1, pois todos os objectos que não foram alvo de rotação utilizam as coordenadas mundo como referencia, e o final, o vector definido pela extrusão, pois esta define qual a direcção para qual o objecto a ser transformado está orientado.

Estes dois vectores, são então normalizados, e depois são aplicadas duas fórmulas, de onde se retira tanto o ângulo como o eixo da rotação aplicando simples operações entre vectores, como o produto interno e externo (Baker, 2008):

- O ângulo é dado pelo  $\arccos$  do produto interno entre os dois vectores normalizados:  
$$a = \cos^{-1}(v_1 \cdot v_2)$$
- O eixo é dado pelo produto externo entre os dois vectores:  $(x, y, z) = v_1 \times v_2$

Esta conversão é a primeira apresentada, onde pode ocorrer perda de informação devido ao processamento de informação necessário para se efectuar a transformação.

- **CmColor → SFCColor**

Nas versões do DWG anteriores à 2004, as cores eram guardadas apenas pelo índice da cor na tabela “AutoCAD Color Index”, sendo depois necessário recorrer a essa tabela para obter o RGB correspondente da cor. A partir da versão 2004, o valor RGB da cor é guardado directamente no ficheiro, tendo-se portanto o valor RGB da cor. Esta conversão é directa para o SFCColor, onde a cor é representada na forma de RGB em três valores de virgula flutuante, apenas com uma nuance, em que o valor RGB tem de ser normalizado para ficar descrito entre o valor 0 e 1.

## B.2. TIPOS COMPOSTOS (OBJECTOS GEOMÉTRICOS)

A um nível mais alto, que o dos tipos, existe a definição dos objectos geométricos que são representados no DWG. Estes objectos irão ter a sua equivalência, com ou sem perda de informação em X3D, para tal faz-se um levantamento dessas equivalências, e aplicam-se as transformações atrás definidas ao nível dos tipos.

De entre vários objectos que o DWG pode representar, foram escolhidos cinco para serem mapeados em X3D: arco, circulo, linha, face tridimensional e sólido.

- **Arco**

Um objecto do tipo arco, é definido por três parâmetros, o raio, o ângulo do início e de fim. Tanto o DWG como o X3D utilizam estes parâmetros da mesma forma, definindo um circulo com o raio definido, desde o ângulo inicial, definido recorrendo-se ao circulo trigonométrico, até ao ângulo final, andando-se no sentido contrario ao ponteiro dos relógios.

---

Como no X3D este arco é centro na origem do referencial e não contem características tridimensionais, é utilizada uma translação e uma rotação a partir da extrusão e o centro definido de forma a levar o arco para o seu local correcto.

Estando o objecto descrito no local correcto, faltam apenas as questões relacionadas com a aparência, a cor e a grossura da linha. Em relação à grossura da linha que descreve o arco, esta é transformada no `linewidthScaleFactor` que define, não a grossura da linha em si, mas uma factor de escala, que resulta numa linha de grossura igual ao valor nominal mais perto do valor definido. O atributo que define a cor no X3D utilizado nesta transformação é o `emissiveColor`, por é o mais directo e que as suas cores não são alteradas consoante as fontes de luz no desenho, permitindo-se obter uma vista mais científica sobre os objectos.

Estes mapeamentos estão descritos na tabela seguinte, e as transformações são obtidas utilizando os mapeamentos de tipos definidos previamente.

TABELA B.1 – MAPEAMENTO DO OBJECTO TIPO ARCO.

DWG (Arc)	X3D (Arc2D)
<ul style="list-style-type: none"> <li>– Color (CmColor)</li> <li>– Center (3DPoint)</li> <li>– Radius (BitDouble)</li> <li>– Thickness (BitThickness)</li> <li>– Extrusion (BitExtrusion)</li> <li>– Start Angle (BitDouble)</li> <li>– End Angle (BitDouble)</li> </ul>	<pre> Transform {   translation (SFVec3f)   rotation (SFRotation)   Shape {     Appearance {       LineProperties {         linewidthScaleFactor (SFFloat)       }       Material {         emissiveColor (SFColor)       }     }     Arc2D {       endAngle (SFFloat)       radius (SFFloat)       startAngle (SFFloat)     }   } } </pre>

- **Circulo**

Este objecto, é análogo ao anterior, inclusive é um caso específico do mesmo, em que o arco é completo, logo, o mapeamento definido é igual ao anterior, mudando o tipo do objecto, Circle2D que apenas, define o raio do circulo.

Na tabela seguinte, é apresentado a equivalência do objecto descrito em DWG no X3D, sendo mais uma vez aplicados os mapeamentos de tipos previamente definidos.

TABELA B.2 – MAPEAMENTO DO OBJECTO TIPO CIRCULO.

DWG (Circle)	X3D (Circle2D)
<ul style="list-style-type: none"> <li>– Color (CmColor)</li> <li>– Center (3DPoint)</li> <li>– Radius (BitDouble)</li> <li>– Thickness (BitThickness)</li> <li>– Extrusion (BitExtrusion)</li> </ul>	<pre> Transform {   translation (SFVec3f)   rotation (SFRotation)   Shape {     Appearance {       LineProperties {         linewidthScaleFactor (SFFloat)       }       Material {         emissiveColor (SFCOLOR)       }     }     Circle2D {       radius (SFFloat)     }   } } </pre>

- **Linha**

A linha, em relação à aparência continua a ir de encontro ao definido nos objectos anteriores, sendo a grande diferença neste caso a inexistência de translações e rotações, isto devido a duas questões. Primeiro, a linha no DWG é definida por dois pontos tridimensionais, e a linha no X3D é definida por duas partes, um vector de pontos tridimensionais, e uma ordem de como esses pontos resultam na linha. Depois há a questão da rotação, que não é aplicada neste caso, visto que o DWG utiliza o campo equivalente, a extrusão, para definir o sentido para onde a grossura da linha cresce.



Devido ao primeiro ponto levantado, é necessário tornar os dois pontos do DWG, o inicial e final, num vector de pontos. Para este vector resultar numa linha, a ordem dos pontos vai ser de um para o outro, visto a linha ser apenas definida com dois pontos. Em relação ao sentido da grossura da linha, esta não é mapeada, porque o X3D não define um sentido específico da grossura, não dando para aplicar esta questão, perdendo-se a informação descrita desta forma.

Na tabela seguinte é possível observar de uma forma mais sucinta o resultado do mapeamento deste tipo.

TABELA B.3 – MAPEAMENTO DO OBJECTO TIPO LINHA.

DWG (Line)	X3D (IndexedLineSet)
<ul style="list-style-type: none"> <li>– Color (CmColor)</li> <li>– Start Point (3DPoint)</li> <li>– End Point (3DPoint)</li> <li>– Thickness (BitThickness)</li> <li>– Extrusion (BitExtrusion)</li> </ul>	<pre>Shape {   Appearance {     LineProperties {       linewidthScaleFactor (SFFloat)     }     Material {       emissiveColor (SFCOLOR)     }   }   IndexedLineSet {     Coordinate {       point (MFVec3f)     }     coordIndex (MFInt32)   } }</pre>

- **Face Tridimensional**

Uma face no espaço tridimensional é descrita no DWG pela cor, e pelos quatro vértices que a compõe, ordenados da forma como é demonstrado na figura seguinte.

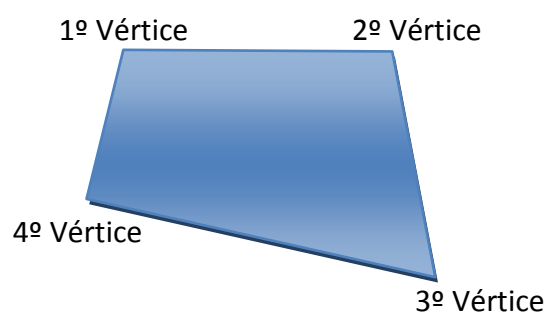


FIGURA B.1 – ORDEM DOS VÉRTICES UTILIZADOS NA DESCRIÇÃO DE UMA FACE TRIDIMENSIONAL.

No X3D tal como na linha, estes pontos são transformados num vector de pontos, que depois são utilizados pela ordem definida. Esta ordem, definida no coordIndex, é a mesma do DWG visto que ambos utilizam a mesma descrição dos sólidos. A cor é transformada da mesma forma como referida anteriormente. Existe agora um campo novo que é a definição do argumento solid, este serve para definir se ambos os lados da face são visíveis ou não. Neste caso este argumento é falso, o que faz com que sejam visíveis, visto que no DWG não é especificado qual o lado visível da face.

Na tabela seguinte é apresentado o resultado deste mapeamento, e como ele resulta no X3D.

TABELA B.4 – MAPEAMENTO DO OBJECTO TIPO FACE TRIDIMENSIONAL.

DWG (3DFace)	X3D (IndexedFaceSet)
<ul style="list-style-type: none"> <li>– Color (CmColor)</li> <li>– 1<sup>st</sup> Corner (3DPoint)</li> <li>– 2<sup>nd</sup> Corner (3DPoint)</li> <li>– 3<sup>rd</sup> Corner (3DPoint)</li> <li>– 4<sup>th</sup> Corner (3DPoint)</li> </ul>	<pre>Shape {   Appearance {     Material {       emissiveColor (SFColor)     }   }   IndexedFaceSet {     Coordinate {       point (MFVec3f)     }     coordIndex (MFInt32)     solid (SFBool)   } }</pre>

- **Sólido**

O DWG define o sólido por uma base, descrita da mesma forma que o objecto anterior com os quatro cantos, por uma grossura, que define a altura do sólido, e por uma extrusão, que define a direcção do mesmo.

No X3D este sólido vai ser definido de uma forma diferente, vai ser definido por seis faces, unindo os oito pontos que definem o sólido, os quatro da base, e os quatro do topo, criados utilizando os da base e somando a grossura definida no DWG. Rodando-se posteriormente o objecto.

Para criar essas seis faces, é necessário definir portanto quais os pontos que elas utilizam como vértices, na figura seguinte é apresentado um exemplo de um sólido, e quais os índices dos diversos vértices que o constituem, sendo depois apresentado qual a ordem que eles são aplicados nas faces de forma a resultar no sólido final.

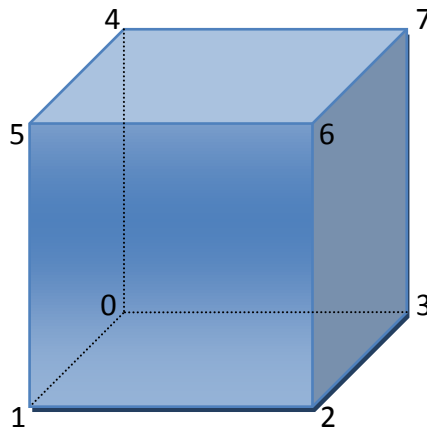


FIGURA B.2 – VERTICES UTILIZADOS PARA DEFINIR AS FACES NO OBJECTO SÓLIDO.

Com o auxílio do desenho, tira-se os vértices constituintes das seis faces, que são:  $0-1-2-3$ ,  $4-5-6-7$ ,  $2-3-7-6$ ,  $0-1-5-4$ ,  $1-2-6-5$  e  $0-3-7-4$ , que resultam no mapeamento definido na tabela seguinte.

TABELA B.5 – MAPEAMENTO DO OBJECTO TIPO SÓLIDO.

DWG (Solid)	X3D (IndexedFaceSet)
<ul style="list-style-type: none"> <li>– Color (CmColor)</li> <li>– Thickness (BitThickness)</li> <li>– 1<sup>st</sup> Corner (3DPoint)</li> <li>– 2<sup>nd</sup> Corner (3DPoint)</li> <li>– 3<sup>rd</sup> Corner (3DPoint)</li> <li>– 4<sup>th</sup> Corner (3DPoint)</li> <li>– Extrusion (BitExtrusion)</li> </ul>	<pre> Transform {   rotation (SFRotation)   Shape {     Appearance {       Material {         emissiveColor (SFColor)       }     }     IndexedFaceSet {       Coordinate {         point (MFVec3f)       }       coordIndex (MFInt32)       solid (SFBool)     }   } } </pre>



## ANEXO C. EXECUÇÃO DOS TESTES

Neste anexo são apresentados, em detalhe, todos os testes executados para a validação da solução. Tal como referido anteriormente, estes testes são divididos em três grupos, cada um correspondente a cada um dos requisitos definidos no desenvolvimento da solução: harmonização dos formatos, personalização de vistas e contextualização da informação.

### C.1. TESTES DA HARMONIZAÇÃO DOS FORMATOS

Com o objectivo de se testar se o mapeamento definido entre o formato DWG e o X3D foi efectuado com sucesso, vai-se conduzir dois testes diferentes: um testando individualmente os elementos geométricos simples definidos, e outro testando um exemplo de um objecto real (semelhante aos utilizados em trocas comerciais).

#### C.1.1. TESTE DE MORFISMO – ELEMENTOS GRÁFICOS SIMPLES

Este teste consiste na verificação do funcionamento dos morfismos entre elementos geométricos simples, e consiste no seguinte:

Teste	
<b>Nome do teste:</b>	Morfismo de elementos simples (Arco, Circulo, Linha, Face 3D e Sólido)
<b>Grupo:</b>	Harmonização dos formatos
<b>Propósito:</b>	Verificar o funcionamento do morfismo entre elementos geométricos simples
<b>Comentários:</b>	O ficheiro DWG utilizado vai ser da versão 2004
Comportamento	Verdicto
! Criar um desenho no AutoCAD com os elementos simples referidos	
! Gravar o desenho obtido no formato DWG versão 2004	
! Abrir o ficheiro obtido na implementação do sistema	
? Ficheiro aberto com sucesso	
! Exportar o ficheiro para X3D	
? Exportação efectuada com sucesso	
! Proceder à comparação entre original e exportado	
? Resultado corresponde ao desenho	Sucesso
? Resultado diferente do desenho	Falha
? Exportação falhada	Falha
? Ficheiro não abriu	Falha

Os dois primeiros passos do teste foram efectuados utilizando a versão de estudante do AutoCAD Electrical 2008 da AutoDesk, e consistiram na criação de um desenho arbitrário que é composto por todos os elementos geométricos simples que contam com morfismos

definidos na implementação. Este desenho foi posteriormente gravado na versão 2004 do DWG de modo a ficar compatível com a prova de conceito definida. Na figura seguinte pode-se observar o desenho criado.

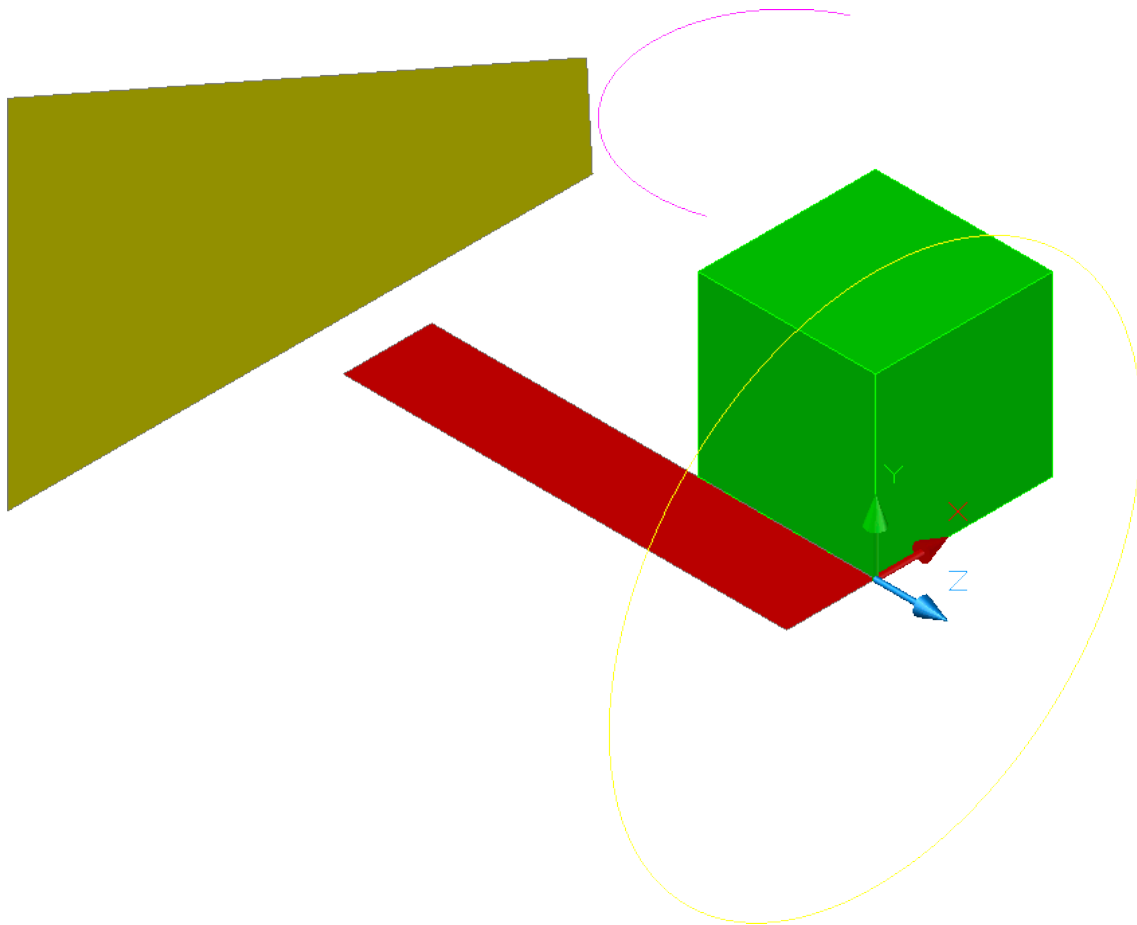


FIGURA C.1 – DESENHO COM ELEMENTOS GEOMETRICOS SIMPLIS A SER TESTADO.

Com o desenho criado, e gravado com sucesso, o próximo passo é passar para abertura do mesmo na prova de conceito. O ficheiro na sua versão 2004 foi aberto com sucesso, logo o próximo passo é a exportação para X3D, que neste caso também ocorre com sucesso.

Tendo então os resultados de exportação pode-se proceder à análise dos resultados, que vai ser efectuada em três fases: uma primeira a validação do ficheiro XML obtido, na segunda efectua-se uma comparação visual do resultado com o original e finalmente faz-se uma análise das características dos objectos definidas no X3D face às originais do AutoCAD.

A primeira fase, passa por uma validação do ficheiro face ao esquema que define o X3D, esta foi efectuada recorrendo-se ao Altova XMLSpy e resultou numa validação com sucesso.

Para a comparação visual é aberto o ficheiro resultante da exportação no visualizador InstantPlayer, que resulta na imagem obtida de seguida.

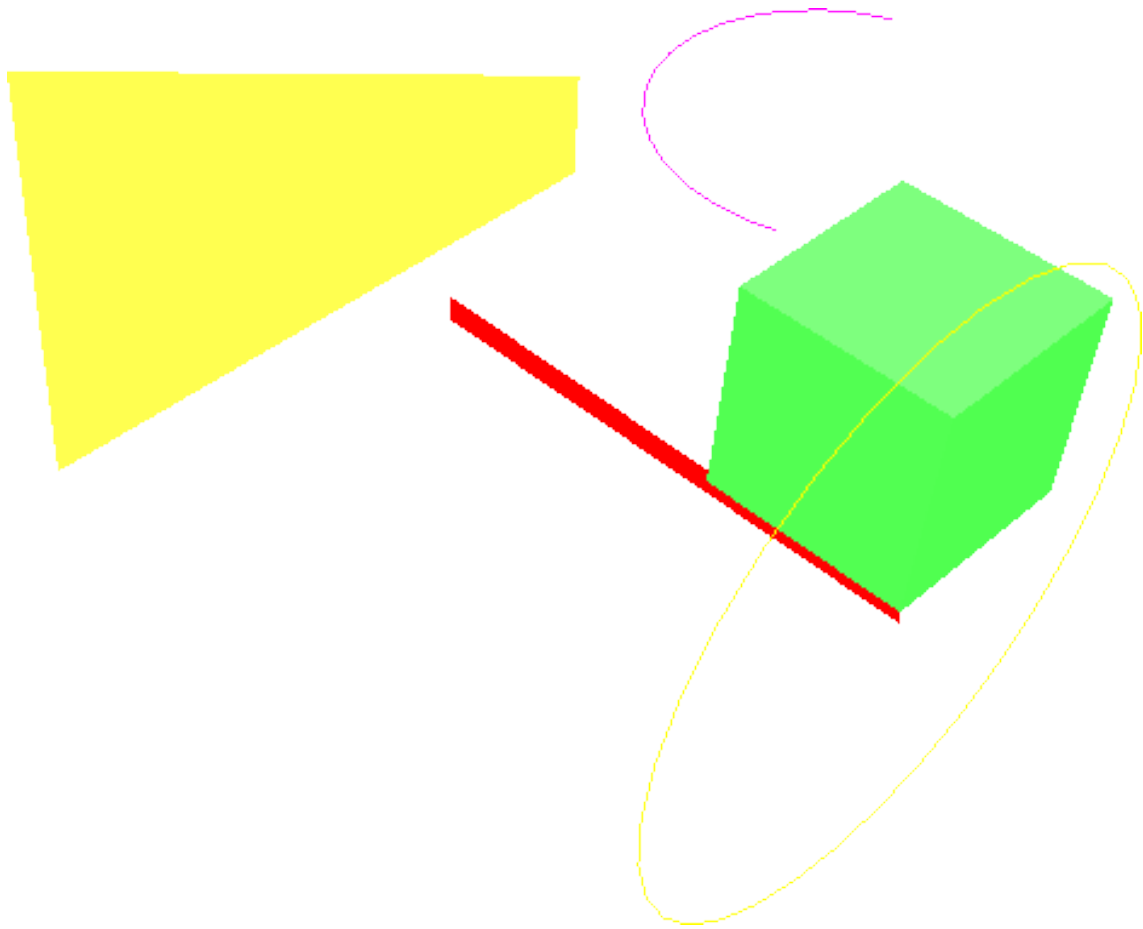


FIGURA C.2 – DESENHO OBTIDO ATRAVÉS DA UTILIZAÇÃO DO X3D EXPORTADO.

Através de uma primeira análise visual, verifica-se que ao nível de escalas, objectos, cores e posições todos os objectos correspondem ao desenho original. Nesta análise a única nuance existente é a orientação da grossura da linha vermelha não estar de acordo com o desenho original, mas esta nuance já havia sido levantada nos morfismos, visto o X3D não definir uma direcção para onde a grossura das linhas deve crescer, logo pela análise visual a exportação foi o esperado.

Feita a análise visual, passa-se para uma análise mais directa aos mapeamentos definidos, verificando-se se as características definidas pelo AutoCAD resultam no X3D da forma definida nos mapeamentos. Seguindo a ordem da definição dos mapeamentos, o primeiro tipo geométrico a ser analisado é o arco, que é descrito no AutoCAD pelas características apresentadas na figura seguinte.

General		Geometry	
Color	■ Magenta	Start X	23.5946
Layer	0	Start Y	26.3393
Linetype	———— ByLayer	Start Z	30
Linetype scale	1	Center X	10
Plot style	ByColor	Center Y	20
Lineweight	———— ByLayer	Center Z	30
Hyperlink		End X	-4.0954
Thickness	0	End Y	14.8697
		End Z	30
		Radius	15
		Start angle	25
		End angle	200
		Total angle	175
		Arc length	45.8149
		Area	333.8067
		Normal X	0
		Normal Y	0
		Normal Z	1

FIGURA C.3 – DEFINIÇÃO DO OBJECTO TIPO ARCO NO AUTOCAD.

O arco no X3D resultou no fragmento do ficheiro apresentado de seguida.

```
<Transform translation="10.00 20.00 30.00" rotation="0.00 0.00 1.00 0.00">
  <Shape>
    <Appearance>
      <LineProperties linewidthScaleFactor="0.0"/>
      <Material emissiveColor="1.00 0.00 1.00"/>
    </Appearance>
    <Arc2D endAngle="3.4906585" startAngle="0.43633232" radius="15.0"/>
  </Shape>
</Transform>
```

FIGURA C.4 – RESULTADO DA EXPORTAÇÃO DO OBJECTO TIPO ARCO PARA X3D.

Para se tirar uma conclusão acerca da equivalência do objecto exportado com o original, é necessário converter-se os valores dos ângulos para radianos, e saber qual o valor da cor utilizada em RGB. O ângulo inicial em radianos equivale a 0.436332313, e o ângulo final a 3.4906585 que corresponde ao descrito no X3D. O magenta, a cor do objecto, corresponde ao RGB de (255,0,255) que normalizado também equivale ao encontrado no X3D.

Tanto o centro, a rotação, a grossura da linha e o raio, cuja conversão é directa, também têm uma correspondência correcta no X3D, o que leva que o resultado da conversão do objecto tipo arco seja sucesso.



O próximo objecto a ser testado é o círculo, e o processo vai assentar no mesmo utilizado para o arco, portanto começa-se por apresentar a informação que descreve o círculo no AutoCAD.

General		Geometry	
Color	<span style="background-color: yellow;">■</span> Yellow	Center X	0
Layer	0	Center Y	0
Linetype	———— ByLayer	Center Z	0
Linetype scale	1	Radius	30
Plot style	ByColor	Diameter	60
Lineweight	———— ByLayer	Circumference	188.4956
Hyperlink		Area	2827.4334
Thickness	0	Normal X	0
		Normal Y	-1
		Normal Z	0

FIGURA C.5 – DEFINIÇÃO DO OBJECTO TIPO CÍRCULO NO AUTOCAD.

Que resulta na seguinte exportação em X3D.

```
<Transform translation="0.00 0.00 0.00" rotation="0.54 0.00 0.00 1.57">
  <Shape>
    <Appearance>
      <LineProperties linewidthScaleFactor="0.0"/>
      <Material emissiveColor="1.00 1.00 0.00"/>
    </Appearance>
    <Circle2D radius="30.0"/>
  </Shape>
</Transform>
```

FIGURA C.6 – RESULTADO DA EXPORTAÇÃO DO OBJECTO TIPO ARCO PARA X3D.

O centro, o raio e a grossura da linha contam com conversão directa, o que se verifica que acontece com sucesso. Mas, para se analisar os restantes campos é necessário o valor RGB correspondente ao amarelo e efectuar processamento dos dados, para se calcular o valor da rotação. O valor RGB do amarelo é (255,255,0), o que normalizado corresponde ao X3D faltando apenas uma conversão correcta da rotação para esta conversão ser dada como um sucesso.

A rotação passa então pelo processo de se calcular qual o ângulo e o eixo de rotação que levam à rotação entre a orientação predefinida do círculo (0,0,1) no X3D e a rotação desejada (0,-1,0). Para tal são aplicadas as fórmulas definidas previamente nos mapeamentos de forma a se tirar o ângulo e o eixo de rotação.

$$(x, y, z) = v_1 \times v_2 \Leftrightarrow \begin{cases} x = y_{v_1} * z_{v_2} - y_{v_2} * z_{v_1} \\ y = z_{v_1} * x_{v_2} - z_{v_2} * x_{v_1} \\ z = x_{v_1} * y_{v_2} - x_{v_2} * y_{v_1} \end{cases} \Leftrightarrow \begin{cases} x = 0 * 0 - (-1) * 1 \\ y = 1 * 0 - 0 * 0 \\ z = 0 * (-1) - 0 * 0 \end{cases} \Leftrightarrow \begin{cases} x = 1 \\ y = 0 \\ z = 0 \end{cases}$$

$$\begin{aligned} \alpha &= \cos^{-1}(v_1 \cdot v_2) = \cos^{-1}(x_{v_1} * x_{v_2} + y_{v_1} * y_{v_2} + z_{v_1} * z_{v_2}) \\ &= \cos^{-1}(0 * 0 + 0 * (-1) + 1 * 0) = \cos^{-1}(0) = 1,570796327 \end{aligned}$$

Os resultados das transformações obtidas, tanto efectuando-se os cálculos manualmente como utilizando a prova de conceito deram resultados diferentes. Apesar de estes terem tido resultados diferentes, este têm o mesmo resultado de aplicação, visto que o tamanho do vector, que é a diferença entre ambos, em nada influencia a rotação, porque este apenas define o sentido de rotação. Logo a exportação de um objecto tipo circulo é um sucesso.

O objecto seguinte é uma linha, que no AutoCAD é definida pelas seguintes características.

General		Geometry	
Color	<span style="color: red;">■</span> Red	Start X	0
Layer	0	Start Y	0
Linetype	———— ByLayer	Start Z	0
Linetype scale	1	End X	0
Plot style	ByColor	End Y	50
Lineweight	———— ByLayer	End Z	0
Hyperlink		Delta X	0
Thickness	10	Delta Y	50
		Delta Z	0
		Length	50
		Angle	90

FIGURA C.7 – DEFINIÇÃO DO OBJECTO TIPO LINHA NO AUTOCAD.

Que exportado para X3D resulta no seguinte objecto.

```
<Shape>
  <Appearance>
    <LineProperties linewidthScaleFactor="10.0"/>
    <Material emissiveColor="1.00 0.00 0.00"/>
  </Appearance>
  <IndexedLineSet coordIndex="0,1">
    <Coordinate point="0.00 0.00 0.00, 0.00 50.00 0.00"/>
  </IndexedLineSet>
</Shape>
```

FIGURA C.8 – RESULTADO DA EXPORTAÇÃO DO OBJECTO TIPO LINHA PARA X3D.

Começando esta análise pela cor, pode-se verificar que a conversão foi efectuada com sucesso, pois o vermelho em RGB é dado por (255,0,0), que normalizado é o que está

presente no X3D. A grossura da linha também é mapeada com sucesso. Nesta transformação existe também algum processamento, visto a linha não ser apenas definida como um início e um fim (o X3D permite definir uma linha com inúmeros pontos), neste caso no vector de pontos presente no X3D ambos os pontos que definem a linha estão representados de forma correcta, tal como a ordem dos mesmos, que corresponde à do ponto inicial para o final. De salientar, o facto de a orientação não ter sido transformada, mas, como já referido nos mapeamentos, o X3D não tem forma de descrever a orientação para onde a grossura da linha cresce, por isso mesmo perdendo-se informação a transformação é um sucesso.

O próximo objecto e o penúltimo é uma face tridimensional, definida no AutoCAD pelas seguintes características.

General		Geometry		Geometry	
Color	Yellow	Vertex	1	Vertex	2
Layer	0	Vertex X	33.1954	Vertex X	-32.7848
Linetype	ByLayer	Vertex Y	65.0553	Vertex Y	65.0553
Linetype scale	1	Vertex Z	-9.63	Vertex Z	-9.5343
Plot style	ByColor	Edge 1	Visible	Edge 1	Visible
Lineweight	ByLayer	Edge 2	Visible	Edge 2	Visible
Hyperlink		Edge 3	Visible	Edge 3	Visible
		Edge 4	Visible	Edge 4	Visible

Geometry		Geometry	
Vertex	3	Vertex	4
Vertex X	-32.7848	Vertex X	32.5208
Vertex Y	65.0553	Vertex Y	65.0553
Vertex Z	30.7844	Vertex Z	2.1126
Edge 1	Visible	Edge 1	Visible
Edge 2	Visible	Edge 2	Visible
Edge 3	Visible	Edge 3	Visible
Edge 4	Visible	Edge 4	Visible

FIGURA C.9 – DEFINIÇÃO DO OBJECTO TIPO FACE TRIDIMENSIONAL NO AUTOCAD.

Que resulta na seguinte descrição em X3D.

```
<Shape>
  <Appearance>
    <Material emissiveColor="1.00 1.00 0.00"/>
  </Appearance>
  <IndexedFaceSet coordIndex="0,1,2,3,-1" solid="false">
    <Coordinate point="33.20 65.06 -9.63, -32.78 65.06 -9.53, -32.78 65.06 30.78, 32.52 65.06 2.11"/>
  </IndexedFaceSet>
</Shape>
```

FIGURA C.10 – RESULTADO DA EXPORTAÇÃO DO OBJECTO FACE TRIDIMENSIONAL PARA X3D.

A face tridimensional apenas conta com a conversão de uma cor, e de os quatro vértices que a compõem. A cor, o amarelo em RGB (255,255,0) tem a conversão bem efectuada. Os

quatro vértices estão descritos correctamente no X3D, e a sua ordem também é a esperada logo esta conversão também é um sucesso.

Por último, a conversão do sólido, que é descrito no AutoCAD pelos seguintes parâmetros.

General		Geometry		Geometry	
Color	Green	Vertex	1	Vertex	2
Layer	0	Vertex X	0	Vertex X	20
Linetype	ByLayer	Vertex Y	0	Vertex Y	0
Linetype scale	1	Elevation	0	Elevation	0
Plot style	ByColor	Geometry		Geometry	
Lineweight	ByLayer	Vertex	3	Vertex	4
Hyperlink		Vertex X	0	Vertex X	20
Thickness	20	Vertex Y	20	Vertex Y	20
		Elevation	0	Elevation	0

FIGURA C.11 – DEFINIÇÃO DO OBJECTO TIPO SÓLIDO NO AUTOCAD.

Que após ser aberto na prova de conceito e exportado para X3D resulta no seguinte.

```
<Transform rotation="0.00 0.00 1.00 0.00">
  <Shape>
    <Appearance>
      <Material emissiveColor="0.00 1.00 0.00"/>
    </Appearance>
    <IndexedFaceSet coordIndex="0,1,2,3,-1,4,5,6,7,-1,2,3,7,6,-1,0,1,5,4,-1,1,2,6,5,-1,0,3,7,4,-1" solid="false">
      <Coordinate point="0.00 0.00 0.00, 20.00 0.00 0.00, 20.00 20.00 0.00, 0.00 20.00 0.00, 0.00 0.00 20.00, 20.00 0.00 20.00, 20.00 20.00 20.00, 0.00 20.00 20.00"/>
    </IndexedFaceSet>
  </Shape>
</Transform>
```

FIGURA C.12 – RESULTADO DA EXPORTAÇÃO DO OBJECTO TIPO SÓLIDO PARA X3D.

Esta conversão é das apresentadas, a que apresenta maior processamento dos dados. Começando pela transformação directa existente, a cor, esta apresenta no X3D o valor esperado. A transformação seguinte, já não é directa como a cor, mas não trás nada de novo, a rotação que neste caso é nenhuma, pois o objecto no AutoCAD situação na origem sem nenhuma rotação aplicada. Em relação à parte específica deste objecto, a definição dos pontos e faces, esta necessita de algum processamento, tanto na criação dos pontos como no seu alinhamento. Como se pode verificar, os pontos presentes no X3D são os pontos definidos individualmente no AutoCAD (que representam a base) e os pontos do topo, criados com a inclusão do parâmetro Thickness aos pontos da base. A ordem dos pontos, descreve as seis faces como de esperado, portanto esta conversão foi também o sucesso.

Após o sucesso da validação, da análise visual, e da análise individual dos objectos cujos morfismos se encontra definidos, pode-se concluir que estes foram definidos de uma forma que permite uma boa transformação entre os formatos.

## C.1.2. TESTE DE MORFISMO – DESENHO COMPOSTO

Este teste consiste na verificação do funcionamento dos morfismos aplicado a um desenho composto, e consiste no seguinte:

<b>Teste</b>	
<b>Nome do teste:</b> Morfismo de um desenho composto	
<b>Grupo:</b> Harmonização dos formatos	
<b>Propósito:</b> Verificar o funcionamento do morfismo na aplicação a um desenho composto	
<b>Comentários:</b> O ficheiro DWG utilizado vai ser da versão 2004	
<b>Comportamento</b>	<b>Verdicto</b>
! Criar um desenho composto no AutoCAD	
! Gravar o desenho obtido no formato DWG versão 2004	
! Abrir o ficheiro obtido na implementação do sistema	
? Ficheiro aberto com sucesso	
! Exportar o ficheiro para X3D	
? Exportação efectuada com sucesso	
! Proceder à comparação entre original e exportado	
? Resultado corresponde ao desenhado	Sucesso
? Resultado diferente do desenhado	Falha
? Exportação falhada	Falha
? Ficheiro não abriu	Falha

Os dois primeiros passos, neste teste fugiram um pouco ao definido, ao invés de se ter criado um desenho, utilizaram-se exemplos reais, provenientes dos casos de uso definidos no projecto onde esta dissertação se inseriu. De seguida é apresentado o desenho utilizado.

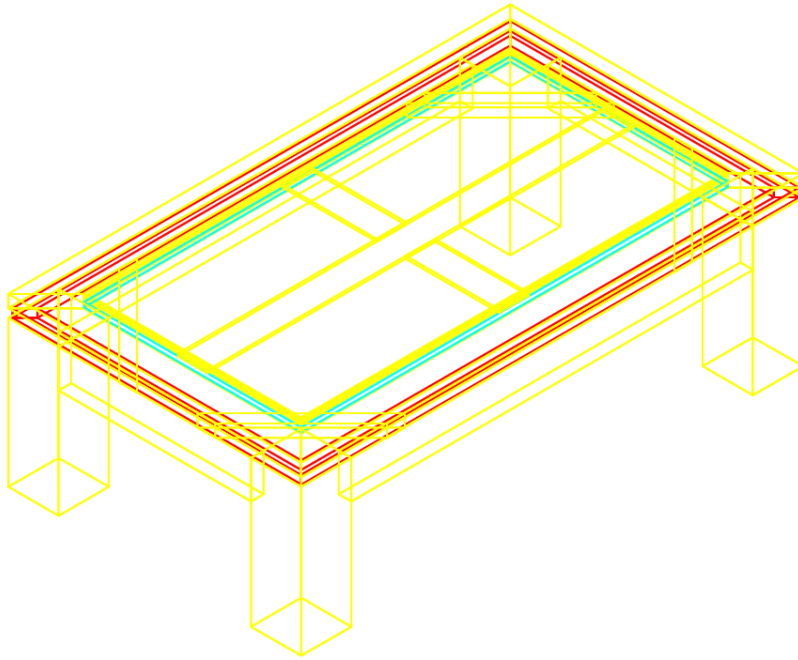


FIGURA C.13 – DESENHO COMPOSTO A SER TESTADO.

Abrindo então o modelo CAD escolhido na prova de conceito, este resulta um ficheiro X3D que foi validado com sucesso, e que aberto no visualizador resulta na seguinte figura:

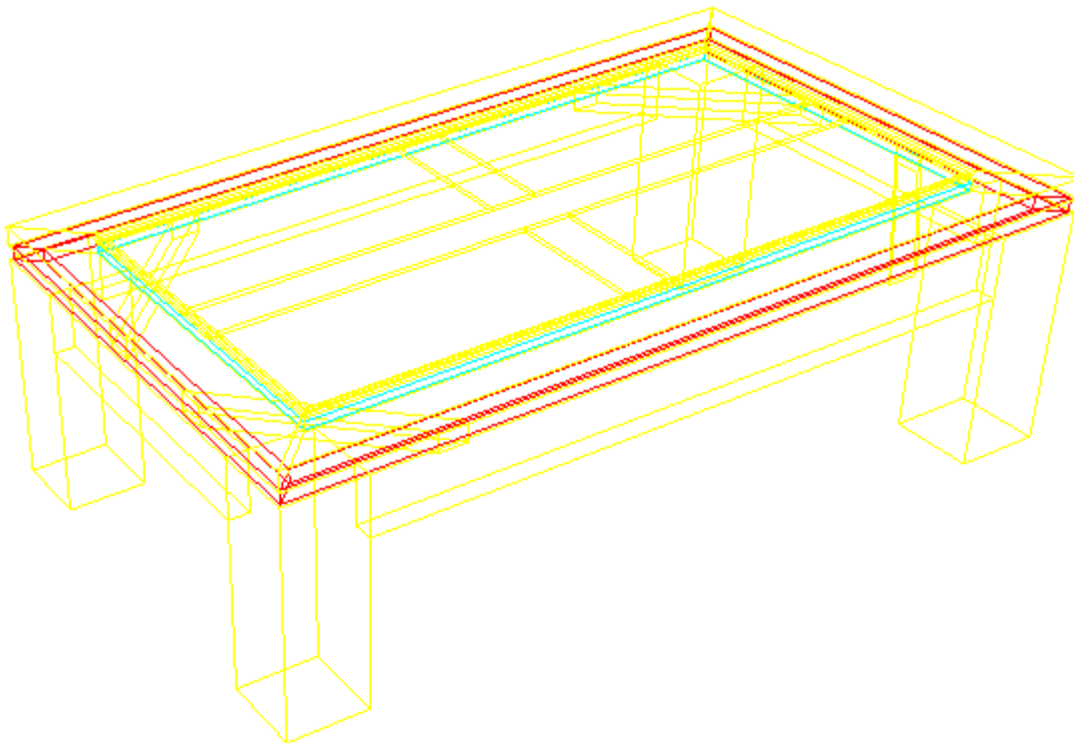


FIGURA C.14 – DESENHO RESULTANTE DA EXPORTAÇÃO.

Analisando portanto o resultado da exportação visualmente, a única utilizada devido a complexidade do ficheiro exportado, pode-se concluir que o desenho corresponde ao original, podendo-se concluir que, ao nível da conversão de formatos, a solução obtém os resultados pretendidos, uma conversão de formatos efectuada com sucesso.

## C.2. TESTES DA PERSONALIZAÇÃO DE VISTAS

Visto a ferramenta implementada contar com a conjugação de dois visualizadores diferentes, vão ser efectuados testes em separado dos mesmos.

### C.2.1. VISUALIZAÇÕES – VISUALIZADOR TRIDIMENSIONAL

Este teste consiste na verificação do funcionamento do visualizador tridimensional incorporado na implementação, e consiste no seguinte:

Teste	
<b>Nome do teste:</b>	Visualizador tridimensional
<b>Grupo:</b>	Personalização de vistas
<b>Propósito:</b>	Verificar o funcionamento do visualizador tridimensional
<b>Comentários:</b>	O ficheiro DWG utilizado vai ser da versão 2004
Comportamento	Verdicto
! Criar um desenho no AutoCAD	
! Gravar o desenho obtido no formato DWG versão 2004	
! Abrir o ficheiro obtido na implementação do sistema	
? Ficheiro aberto com sucesso	
? Desenho no visualizador correspondente ao desenhado	Sucesso
? Desenho no visualizador sem correspondência ao desenhado	Falha
? Ficheiro não abriu	Falha

Os três passos iniciais foram efectuados de forma análoga ao “*Teste de morfismo – Elementos gráficos simples*”, sendo inclusive utilizado o mesmo ficheiro de testes.

Após a abertura do ficheiro, a visualização no visualizador tridimensional foi falhada, o que depois de uma análise inicial, se verificou que o problema se deveu a que o visualizador incorporado na implementação não suportar objectos do tipo círculo, como tal, levou à falha total da apresentação do desenho. Levando a que este teste tenha sido falhado.

### C.2.2. VISUALIZAÇÕES – VISTA TABULAR

Este teste consiste na verificação do funcionamento da vista tabular incorporada na implementação, e consiste no seguinte:

Teste	
<b>Nome do teste:</b>	Vista tabular
<b>Grupo:</b>	Personalização de vistas
<b>Propósito:</b>	Verificar o funcionamento da vista tabular
<b>Comentários:</b>	O ficheiro DWG utilizado vai ser da versão 2004
Comportamento	Veredicto
! Criar um desenho no AutoCAD com objectos de cores e materiais diferentes	
! Gravar o desenho obtido no formato DWG versão 2004	
! Abrir o ficheiro obtido na implementação do sistema	
? Ficheiro aberto com sucesso	
? Materiais e cores da tabela correspondem com as desenhadas	Sucesso
? Tabela com informação errada	Falha
? Ficheiro não abriu	Falha

Os três passos iniciais foram novamente efectuados de forma análoga ao “*Teste de morfismo – Elementos gráficos simples*”, sendo novamente utilizado o mesmo ficheiro de testes. Sendo que desta feita, o mesmo foi alterado para conter definições de materiais.

Foram definidos cinco tipos diferentes de materiais, que se encontram em concordância com os objectos geométricos utilizados, portanto foram definidos o “Material Cubo”, o “Material Circulo”, “Material Plano”, “Material Arco” e “Material Linha” que foram associados aos objectos respectivos.

Aplicando então o ficheiro criado à prova de conceito, obtém-se a seguinte vista tabular.

Color	Part Name	Material	Texture	Description
Green		Material Cubo		
Yellow		Material Circulo		
Magenta		Material Arco		
Red		Material Linha		

FIGURA C.15 – VISTA TABULAR OBTIDA.

Analisando o resultado obtido, verifica-se que só quatro dos cinco materiais são apresentados. Após uma breve análise conclui-se que tal se deveu ao facto de existirem dois materiais que estão associados à mesma cor, e como a tabela está definida para não apresentar cores repetidas apenas um dos materiais apareceu associado à cor amarela.



Apesar de se poder concluir que este visualizador funciona em conformidade com o esperado, pode-se também concluir que a utilização desta vista pode levar à perda de informação.

### C.3. TESTES DA CONTEXTUALIZAÇÃO DA INFORMAÇÃO

Ao invés dos testes anteriores, este não contem sub-testes, contendo apenas um único teste sobre a contextualização da informação, que passa pelo agrupamento de objectos pelas suas cores.

Teste	
<b>Nome do teste:</b>	Agrupamento dos objectos por cor
<b>Grupo:</b>	Contextualização da informação
<b>Propósito:</b>	Verificar o funcionamento da contextualização da informação
<b>Comentários:</b>	O ficheiro DWG utilizado vai ser da versão 2004
Comportamento	Veredicto
! Criar um desenho no AutoCAD com objectos de cores e materiais diferentes	
! Gravar o desenho obtido no formato DWG versão 2004	
! Abrir o ficheiro obtido na implementação do sistema	
? Ficheiro aberto com sucesso	
! Exportar com e sem a aplicação de regras o ficheiro para X3D	
? Exportação efectuada com sucesso	
! Comparação de ambos com a definição da regra	
? Os resultados de ambos foram diferentes	
? Regra aplicada com sucesso	Sucesso
? Regra sem efeitos	Falha
? Ambos tiveram o mesmo resultado	Falha
? Exportação falhada	Falha
? Ficheiro não abriu	Falha

Mais uma vez o ficheiro de testes a ser utilizado é um ficheiro previamente criado e utilizado, desta feita no “*Visualizações – Vista Tabular*”. Este teste numa primeira fase passa pela exportação do desenho sem a aplicação da contextualização da informação, o que resulta na seguinte saída.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<X3D xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" profile="Immersive"
version="3.0"
xsi:noNamespaceSchemaLocation="http://www.web3d.org/specifications/x3d-3.2.xsd">
  <head>
    <meta content="Simples - Materiais.x3d" name="Title"/>
    <meta content="Mon Feb 02 16:37:13 GMT 2009" name="Created"/>
    <meta content="UNINOVA-GRIS GeODED Visualizer - Open Visualizer for
Geometry Data Encoded in DWG Format" name="Generator"/>
  </head>
```

```

<Scene>
  <Transform rotation="0.00 0.00 1.00 0.00">
    <Shape>
      <Appearance>
        <Material emissiveColor="0.00 1.00 0.00"
DEF="Color_green__Material_Cubo">
          <MetadataString value="Material Cubo" name="Material"/>
        </Material>
      </Appearance>
      <IndexedFaceSet coordIndex="0,1,2,3,-1,4,5,6,7,-1,2,3,7,6,-
1,0,1,5,4,-1,1,2,6,5,-1,0,3,7,4,-1" solid="false">
        <Coordinate point="0.00 0.00 0.00, 20.00 0.00 0.00, 20.00 20.00
0.00, 0.00 20.00 0.00, 0.00 0.00 20.00, 20.00 0.00 20.00, 20.00 20.00 20.00, 0.00
20.00 20.00"/>
      </IndexedFaceSet>
    </Shape>
  </Transform>
  <Transform translation="10.00 20.00 30.00" rotation="0.00 0.00 1.00 0.00">
    <Shape>
      <Appearance>
        <LineProperties linewidthScaleFactor="0.0"/>
        <Material emissiveColor="1.00 0.00 1.00"
DEF="Color_magenta__Material_Arco">
          <MetadataString value="Material Arco" name="Material"/>
        </Material>
      </Appearance>
      <Arc2D endAngle="3.4906585" startAngle="0.43633232" radius="15.0"/>
    </Shape>
  </Transform>
  <Transform translation="0.00 0.00 0.00" rotation="-1.00 -0.00 0.00 1.57">
    <Shape>
      <Appearance>
        <LineProperties linewidthScaleFactor="0.0"/>
        <Material emissiveColor="1.00 1.00 0.00"
DEF="Color_yellow__Material_Circulo">
          <MetadataString value="Material Circulo" name="Material"/>
        </Material>
      </Appearance>
      <Circle2D radius="30.0"/>
    </Shape>
  </Transform>
  <Shape>
    <Appearance>
      <Material emissiveColor="1.00 1.00 0.00"
DEF="Color_yellow__Material_Plano">
        <MetadataString value="Material Plano" name="Material"/>
      </Material>
    </Appearance>
    <IndexedFaceSet coordIndex="0,1,2,3,-1" solid="false">
      <Coordinate point="33.20 65.06 -9.63, -32.78 65.06 -9.53, -32.78
65.06 30.78, 32.52 65.06 2.11"/>
    </IndexedFaceSet>
  </Shape>
  <Shape>
    <Appearance>
      <LineProperties linewidthScaleFactor="10.0"/>
      <Material emissiveColor="1.00 0.00 0.00"
DEF="Color_red__Material_Linha">
        <MetadataString value="Material Linha" name="Material"/>
      </Material>
    </Appearance>
    <IndexedLineSet coordIndex="0,1">
      <Coordinate point="0.00 0.00 0.00, 0.00 50.00 0.00"/>
    </IndexedLineSet>
  </Shape>
</Scene>
</X3D>

```

FIGURA C.16 – EXPORTAÇÃO DO FICHEIRO SEM APLICAÇÃO DE REGRAS DE CONTEXTUALIZAÇÃO.

Portanto, tendo já o ficheiro exportado sem regras, pode-se proceder à mesma exportação mas desta vez com regras, para posteriormente se poder efectuar a comparação. O resultado obtido da exportação foi o seguinte.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<X3D xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" profile="Immersive"
version="3.0"
xsi:noNamespaceSchemaLocation="http://www.web3d.org/specifications/x3d-3.2.xsd">
  <head>
    <meta content="Simples - Materiais - Contexto.x3d" name="Title"/>
    <meta content="Mon Feb 02 16:37:16 GMT 2009" name="Created"/>
    <meta content="UNINOVA-GRIS GeODED Visualizer - Open Visualizer for
Geometry Data Encoded in DWG Format" name="Generator"/>
  </head>
  <Scene>
    <Group DEF="Group_green">
      <WorldInfo title="Material" info="Material Cubo"/>
      <Transform rotation="0.00 0.00 1.00 0.00">
        <Shape>
          <Appearance>
            <Material emissiveColor="0.00 1.00 0.00"
DEF="Color_green_Material_Cubo">
              <MetadataString value="Material Cubo" name="Material"/>
            </Material>
          </Appearance>
          <IndexedFaceSet coordIndex="0,1,2,3,-1,4,5,6,7,-1,2,3,7,6,-
1,0,1,5,4,-1,1,2,6,5,-1,0,3,7,4,-1" solid="false">
            <Coordinate point="0.00 0.00 0.00, 20.00 0.00 0.00, 20.00
20.00 0.00, 0.00 20.00 0.00, 0.00 0.00 20.00, 20.00 0.00 20.00, 20.00
0.00 20.00"/>
          </IndexedFaceSet>
        </Shape>
      </Transform>
    </Group>
    <Group DEF="Group_magenta">
      <WorldInfo title="Material" info="Material Arco"/>
      <Transform translation="10.00 20.00 30.00" rotation="0.00 0.00 1.00
0.00">
        <Shape>
          <Appearance>
            <LineProperties linewidthScaleFactor="0.0"/>
            <Material emissiveColor="1.00 0.00 1.00"
DEF="Color_magenta_Material_Arco">
              <MetadataString value="Material Arco" name="Material"/>
            </Material>
          </Appearance>
          <Arc2D endAngle="3.4906585" startAngle="0.4363232"
radius="15.0"/>
        </Shape>
      </Transform>
    </Group>
    <Group DEF="Group_yellow">
      <WorldInfo title="Material" info="Material Circulo"/>
      <Transform translation="0.00 0.00 0.00" rotation="-1.00 -0.00 0.00
1.57">
        <Shape>
          <Appearance>
            <LineProperties linewidthScaleFactor="0.0"/>
            <Material emissiveColor="1.00 1.00 0.00"
DEF="Color_yellow_Material_Circulo">
              <MetadataString value="Material Circulo"
name="Material"/>
            </Material>
          </Appearance>
          <Circle2D radius="30.0"/>
        </Shape>
      </Transform>
    </Group>
  </Scene>
</X3D>
```

```

    <Shape>
      <Appearance>
        <Material emissiveColor="1.00 1.00 0.00"
DEF="Color_yellow__Material_Plano">
          <MetadataString value="Material Plano" name="Material"/>
        </Material>
      </Appearance>
      <IndexedFaceSet coordIndex="0,1,2,3,-1" solid="false">
        <Coordinate point="33.20 65.06 -9.63, -32.78 65.06 -9.53, -
32.78 65.06 30.78, 32.52 65.06 2.11"/>
      </IndexedFaceSet>
    </Shape>
  </Group>
  <Group DEF="Group_red">
    <WorldInfo title="Material" info="Material Linha"/>
    <Shape>
      <Appearance>
        <LineProperties linewidthScaleFactor="10.0"/>
        <Material emissiveColor="1.00 0.00 0.00"
DEF="Color_red__Material_Linha">
          <MetadataString value="Material Linha" name="Material"/>
        </Material>
      </Appearance>
      <IndexedLineSet coordIndex="0,1">
        <Coordinate point="0.00 0.00 0.00, 0.00 50.00 0.00"/>
      </IndexedLineSet>
    </Shape>
  </Group>
</Scene>
</X3D>

```

FIGURA C.17 – EXPORTAÇÃO DO FICHEIRO COM APLICAÇÃO DE REGRAS DE CONTEXTUALIZAÇÃO.

Analisando os resultados obtidos, verifica-se que a aplicação da regra contextual definida foi um sucesso. Apenas de notar que, visto o nome do grupo ser definido pelo material do primeiro objecto do grupo (de forma análoga à vista tabular), que esse nome não se encontra totalmente correcto, pois existem diversos materiais dentro do grupo amarelo, o que não invalida o agrupamento com sucesso dos objectos por cores.