

Modelação Expedita de Edifícios Monumentais a Partir de Descrições Textuais

Roberto Rodrigues¹

António Coelho^{1,2}

Luís Paulo Reis^{1,3}

¹ DEI/FEUP, Rua Dr. Roberto Frias, s/n, 4200-465 Porto, Portugal

² INESC Porto, Campus da FEUP, Rua Dr. Roberto Frias, s/n, 4200-465 Porto, Portugal

³ LIACC, Rua Dr. Roberto Frias, s/n, 4200-465 Porto, Portugal
{pro06010, acoelho, lpreis}@fe.up.pt

Resumo

A criação de modelos tridimensionais de ambientes urbanos utilizando modelação expedita é apresentada como sendo uma solução que permite ganhos financeiros e temporais, mantendo um nível de fidelidade visual aceitável. No entanto, a modelação de edifícios âncora (ou monumentais), identificativos de determinadas zonas urbanas, necessita de uma modelação mais cuidada devido ao elevado nível de detalhe necessário, recorrendo-se, geralmente, à sua modelação manual.

Este artigo apresenta uma proposta de automatização do processo de modelação de edifícios através da introdução de conhecimento adicional num sistema de modelação expedita, a partir de descrições textuais. Embora os resultados obtidos não apresentem um nível de fidelidade visual muito elevada, exibem elevados níveis de correspondência entre o texto introduzido e o modelo tridimensional criado. Os resultados obtidos demonstram que a estrutura interna do protótipo é suficientemente flexível para construir distintos modelos de igreja e outros edifícios monumentais, com um nível de realismo muito aceitável. Os resultados demonstram ainda que é possível criar um modelo 3D a partir de um texto escrito em português permitindo, deste modo, que utilizadores não especializados possam utilizar o sistema, reduzindo os custos globais de realização do projecto.

Palavras-chave

Modelação expedita, processamento de linguagem natural.

1. INTRODUÇÃO

Assistimos, cada vez mais, à utilização de Realidade Virtual nas mais diversas áreas, desde as aplicações de carácter mais lúdico, como os videojogos, até às utilizações mais técnicas como a pré-visualização de estruturas que ainda não foram construídas. Os avanços tecnológicos que permitiram um aumento da capacidade de processamento dos sistemas informáticos, associados à utilização de arquitecturas mais eficientes, levaram à vulgarização da sua utilização potenciando assim um aumento de competitividade entre as empresas, e levando a uma melhoria dos serviços oferecidos e à optimização dos custos.

Nos ambientes urbanos, estas aplicações podem ser utilizadas em áreas como o turismo e o marketing, planeamento urbano, resposta a situações de emergência ou mesmo em estudos de impacto ambiental e propagação de ruído.

Embora existam diversas ferramentas interactivas para a reconstrução 3D de edifícios, estas exigem mão-de-obra especializada e grandes recursos informáticos. A criação de uma ferramenta de modelação expedita de edifícios urbanos, que possa utilizar informação armazenada e conhecimento sobre a estrutura a modelar, permitiria desta forma, uma grande redução dos custos, quer pela não

utilização de mão-de-obra especializada, quer pela redução do tempo dispendido durante a execução da mesma.

Entre os edifícios mais representativos de um ambiente urbano, e em especial das cidades, estão os seus edifícios âncora, identificativos de determinadas zonas de cidades e facilmente reconhecíveis, quer pelos seus habitantes, quer por forasteiros. Entre estes, os monumentos serão porventura aqueles que necessitarão de maior detalhe, de forma a serem facilmente reconhecidos, aumentando o tempo de modelação, sendo por isso mais dispendiosos. Embora a modelação expedita permita ganhos de tempo e redução de custos, é generalista, sendo necessário, no caso de edifícios muito detalhados, complementar os modelos criados através de modelação manual.

Por outro lado, existe um vasto conjunto de elementos textuais descritivos de monumentos, quer em livros, quer em instituições como as câmaras municipais ou o Instituto Português do Património Arquitectónico, que contém muita informação que pode ser utilizada para adicionar, de forma automática ou assistida, detalhe aos modelos 3D desses edifícios.

Apresentamos assim uma proposta para a modelação expedita de edifícios a partir de texto descritivo. O texto descritivo é convertido para um formato intermédio em

XML, sendo de seguida convertido para cityGML [Kolbe05], podendo este modelo 3D ser visualizado imediatamente através dos visualizadores existente.

O artigo está estruturado da seguinte forma: na secção 2 será apresentado algum do trabalho realizado nesta área, sendo mostrada a arquitectura proposta na secção 3. Na secção 4 serão apresentados alguns dos resultados obtidos e na secção 5 serão apresentadas as conclusões e as propostas de desenvolvimentos futuros.

2. TRABALHO RELACIONADO

2.1 Modelação tridimensional de edifícios e ambientes urbanos

A criação de modelos tridimensionais é um processo moroso e dispendioso e, apesar de existirem alguns casos que tenham sido criados por modelação manual e individual de cada um dos seus componentes, os processos semi-automáticos provaram que podem ser muito mais eficientes, reduzindo a duração, o custo e a alocação de recursos humanos [Pimentel01].

Podem ser utilizados na arqueologia, para a visualização de edifícios ou cidades que já não existem ou que foram destruídos, existindo inúmeros exemplos de projectos nacionais e internacionais nessa área. Entre outros projectos nacionais, destaca-se a reconstrução de Bracara Augusta [Bernardes03]. Internacionalmente destaca-se o trabalho realizado no Sítio Romano de Sagalassos no Sudoeste da Turquia [Pollefeys00], a reconstrução da cidade medieval Turku na Finlândia [Uotila00] e mais recentemente o projecto “Rome Reborn” [Frischer08], que foi apresentado no SIGGRAPH 2008 na secção de “New Tech Demos”.

Podem também ser utilizados para a visualização de cidades actuais, ou em simulações ambientais como mapas de ruído, simuladores de treino ou de catástrofes, na arquitectura ou no planeamento urbano [Shiode01] [Döllner06]. Na avaliação do impacto de projectos urbanísticos, os modelos tridimensionais apresentam-se como um recurso poderoso no design e planeamento urbano pois permitem avaliar qual seria o impacto paisagístico das novas edificações e intervenções. As aplicações interactivas que permitam uma manipulação dos modelos e uma automatização do processo de modelação são por isso áreas de investigação importantes. O software Polytrim [Danahy95], desenvolvido pela CLR (Centre for Landscape Research) da Universidade de Toronto foi uma das iniciativas pioneiras. Em Portugal, o projecto Lisboa 3D [Pimentel01] serve de base à gestão urbanística da cidade de Lisboa, podendo ser utilizado, por exemplo, para avaliar qual o impacto visual de novos edifícios.

A maioria dos projectos utiliza como base tecnológica o VRML visto que a grande maioria dos modeladores comerciais permite a exportação para este formato, existindo também um elevado número de visualizadores gratuitos para as mais diversas plataformas. Com o advento do X3D este formato deverá começar a ser cada vez mais

utilizado, originando a migração para este formato [Anslow06].

A proposta cityGML [Kolbe05], adoptada em Agosto de 2008 como um standard oficial do Open Geospatial Consortium, deu origem a inúmeros projectos comerciais e de investigação, sendo o seu expoente o 3D Geo e o seu editor LandXPlorer desenvolvido pela 3D Geo e pelo Hasso-Plattner-Institute da Universidade de Potsdam. A nova versão deste software foi adquirida pela Autodesk, fazendo desta forma parte da gama de produtos comerciais que a Autodesk disponibiliza para modelação 3D.

2.2 Modelação expedita

Os custos e o tempo utilizado para a modelação de ambientes urbanos levaram ao surgimento de diversos trabalhos de investigação na área da automatização desses processos. Esses trabalhos estão direccionados para a criação de soluções expeditas que permitam minimizar a intervenção humana e que demorem o menor tempo possível.

Existem duas linhas orientadoras para este processo:

- Automatização do processo de recolha e interpretação de dados;
- Ampliação dos dados originais através de uma base alargada de conhecimento.

No primeiro caso, é necessário recolher um conjunto alargado de informações sobre o ambiente a modelar que será processado por algoritmos de interpretação para dar origem a informação estruturada. Desta forma será possível automatizar a reconstrução da versão tridimensional do ambiente urbano. Além de fotografias e vídeo, as malhas de pontos tridimensionais, as tecnologias de varrimento por LASER ou raios X são também utilizadas.

No segundo caso, a ampliação dos dados originais é obtida através de ferramentas matemáticas que ampliam a informação original. As ferramentas devem incorporar o conhecimento do processo de modelação tal como foi especificado pelo utilizador. Os sistemas L e as gramáticas de cisão, estão entre as ferramentas que permitem um elevado grau de ampliação dos dados.

Com a proliferação de sistemas informáticos a informação passou a estar disponível em fontes diversas e muitas vezes não compatíveis. Uma proposta de modelação expedita semi-automática [Coelho07] propõe um sistema para modelação de ambientes urbanos, com acesso a diferentes fontes de informação, e que utiliza sistemas L para efectuar a modelação.

Os trabalhos de Wonka e Müller, relativamente à utilização das gramáticas de cisão [Wonka03] para criarem fachadas detalhadas [Mueller07] e edifícios completos [Mueller06] permitiram a criação do ambiente de desenvolvimento City Engine, que se apresenta como uma solução eficiente para a modelação 3D de ambientes urbanos.

Com o advento da Web Semântica, que se apresenta como uma evolução da World Wide Web, onde se pre-

tende que a informação disponível também seja compreendida pelas máquinas, surgiram novas tecnologias e novas especificações. Pretende-se desta forma que exista uma especificação de conceitos de forma a eliminar a ambiguidade permitindo que as máquinas respondam e compreendam as solicitações de um utilizador humano. O XML surge como uma forma de representação da informação em ficheiros de texto, permitindo uma troca interoperável de informação entre sistemas distintos.

No âmbito da computação gráfica, foi apresentado um sistema [Liu06] para a reconstrução de habitações chinesas antigas, utilizando um modelo semântico. Este sistema converte os componentes geométricos como pontos, linhas e triângulos em componentes semânticas, como ruas, blocos e casas.

2.3 Ontologia para Computação Gráfica

Utilizado inicialmente em Filosofia, o termo ontologia diz respeito à Natureza e à organização do Ser. Em Ciências da Informação, uma ontologia é um documento ou arquivo que formalmente define as relações entre os termos. O caso mais típico de ontologia para a Internet tem uma taxonomia e um conjunto de regras de inferência [Berners-Lee01]. A taxonomia define as classes de objectos e as relações entre eles.

As ontologias permitem assim [Guimarães02]:

- A existência de um vocabulário para o Conhecimento, de forma a eliminar as ambiguidades;
- A partilha do Conhecimento. Ao definir um vocabulário para um dado Domínio, diversas entidades podem passar a utilizar esse vocabulário para desenvolver aplicações para esse Domínio.
- Uma descrição exacta do Conhecimento. Ao contrário da linguagem natural, onde uma dada palavra poderá ter significados ambíguos, dependentes do contexto, as ontologias descrevem os conceitos em linguagem formal e desta forma a interpretação de uma dada palavra está sempre contextualizada pelo Domínio que está a ser utilizado pelos intervenientes.
- Representação em multi-linguagens, pois é possível fazer o mapeamento de uma ontologia em diversas línguas.
- Criar uma ontologia para um Domínio específico estendendo uma ontologia mais genérica já existente.

Dos diversos formatos existentes para a representação dos modelos 3D, o cityGML, baseado no standard GML3 do Open Geospatial Consortium, cobre os aspectos geométricos, topológicos e semânticos de modelos urbanos tridimensionais. Permite distinguir edifícios de outros artefactos feitos pelo homem, como sejam vegetação, rios, ruas e pistas. Está estruturado de forma a permitir 5 níveis de detalhe: desde o nível 0, que representa o modelo digital do terreno, até ao nível 4, que representa o interior do edifício.

É um formato livre e baseado em standards ISO, podendo ser especializado ou generalizado de acordo com as

necessidades, e suporta geometria e topologia 3D simples ou complexa. Armazena não só a informação geométrica, referente à forma e características topologia dos edifícios, como também armazena informação semântica, existindo uma correspondência entre o objecto geométrico e o objecto semântico.

2.4 Processamento de linguagem natural

O processamento de linguagem natural é uma área de investigação da inteligência artificial que, entre outros objectivos, tenta converter a linguagem humana em representações formais que os computadores sejam capazes de manipular, tendo sido desenvolvidos e apresentados muitos projectos ao longo dos anos. Entre as suas inúmeras aplicações encontramos os componentes dos motores de busca e as ferramentas automáticas de tradução e geração de resumos [Baeza-Yates04].

A plataforma Gate - General Architecture for Text Engineering e o projecto NLTK - Natural Language Toolkit são algumas das soluções desenvolvidas que permitem auxiliar os investigadores e programadores que desenvolvam aplicações que necessitem de algum tipo de processamento de linguagem natural. Enquanto o NLTK se apresenta como um conjunto de bibliotecas e programas desenvolvidos em Python [Garrette09], a plataforma Gate apresenta-se como uma infra-estrutura open-source de desenvolvimento de componentes para processamento de linguagem natural [Bontcheva02].

A plataforma NooJ [Silberstein04] é um ambiente de desenvolvimento linguístico colaborativo e gratuito que inclui um vasto conjunto de recursos e módulos em mais de uma dezena de línguas. Com esta plataforma é possível analisar um texto, ou conjunto de textos, relativamente à sua estrutura gramatical, sintáctica ou morfológica podendo processar mais de 100 formatos de ficheiros, incluindo PDF e HTML. Por outro lado, o sistema de anotações do NooJ não altera o texto original, permitindo por exemplo que sejam aplicadas várias gramáticas em cascata, sendo que as anotações podem ser exportadas directamente para documentos XML. Para a criação das gramáticas o NooJ disponibiliza uma ferramenta gráfica que facilita a criação e manutenção das gramáticas desenvolvidas. Estas gramáticas podem também ser reutilizadas na criação de novas gramáticas. Apesar de só funcionar em sistemas Windows, os recursos desenvolvidos podem ser incorporadas numa ferramenta Web permitindo desta forma a sua utilização em outros sistemas operativos.

3. MODELAÇÃO EXPEDITA DE EDIFÍCIOS A PARTIR DE INFORMAÇÃO TEXTUAL

Os edifícios monumentais são constituídos por estruturas complexas como por exemplo catedrais, palácios e castelos, que necessitam de um maior pormenor na modelação dos detalhes. O desenvolvimento de um protótipo que permita extrair toda a informação relevante de descrições textuais para de seguida criar um modelo 3D com elevada fidelidade visual é um processo complexo que foi dividido em diversas fases de desenvolvimento.

Na primeira, pretende-se gerar um modelo 3D realista, que represente um edifício descrito em português, utilizando uma gramática simples, mas representativa da estrutura. Optou-se nesta fase por gerar um edifício com uma estrutura mais simples mas que fizesse parte do grupo de edifícios monumentais, as igrejas.

Assim, o processo de modelação proposto vai gerar um modelo 3D de uma igreja, no formato cityGML a partir de um texto em linguagem natural, escrito em português.

A figura 1 apresenta o esquema da arquitectura geral do processo de modelação.

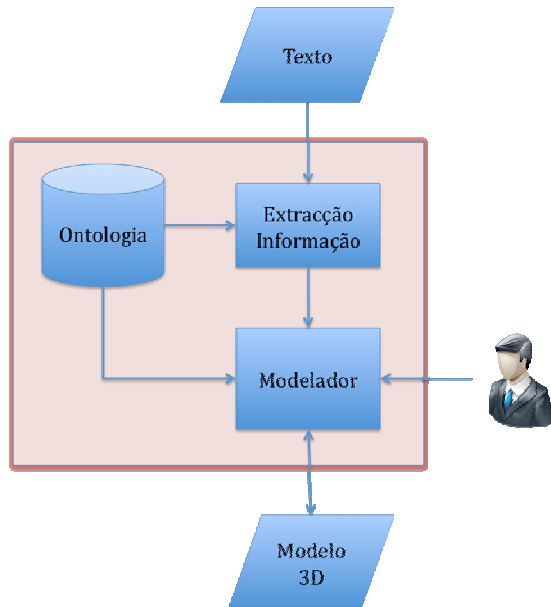


Figura 1: Arquitectura geral

O processo de modelação divide-se em duas fases:

- Extração das características do modelo, contidas no ficheiro de texto escrito em linguagem natural, para um formato interoperável e armazenado na forma de um ficheiro XML. Este ficheiro deve obedecer às especificações do esquema XML do qual se apresenta um extracto na figura 2;

```

<xs:element name="edifício">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="tipo" />
      <xs:element ref="nome" />
      <xs:element ref="corpoPrincipal" />
      <xs:element ref="torres" />
      <xs:element ref="entradas" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
  
```

Figura 2: Extracto do esquema XML do modelo interno

- Conversão do modelo interno para um modelo 3D, no formato cityGML.

Nesta fase de desenvolvimento do protótipo, é necessário colocar dados quantitativos relativamente às características do edifício, nomeadamente as suas dimensões.

Relativamente à arquitectura inicialmente proposta foram efectuadas algumas simplificações, que são apresentadas de seguida.

3.1 Extração de informação

A figura 3 mostra a arquitectura do módulo de extração de informação, desenvolvido utilizando o Nooj. Este módulo recebe um texto em linguagem natural, escrito em português, e converte-o para o modelo interno, armazenando-o num documento XML.

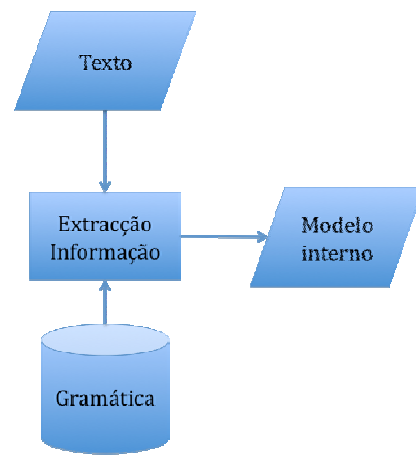


Figura 3: Módulo de extração de informação

Esta gramática utiliza os recursos de reconhecimento de português implementados pelo portal Port4Nooj [Ribeiro08], tendo sido desenvolvidos recursos específicos para esta área.

Foram analisados os formatos de igrejas e os seus textos descritivos, tendo sido criado um texto típico para este tipo de edifício que o descreva de forma realista.

Assim, o texto deverá indicar qual é o tipo de edifício que está a ser analisado, indicando qual a sua constituição em termos de edifícios ou abas. O texto deve também indicar explicitamente a localização do edifício principal e a posição relativas das abas ou torres. Para cada edifício, aba ou torre deve também ser indicada explicitamente qual a sua dimensão e qual o tipo de telhado.

Uma igreja pode então ser descrita por um texto como o representado na figura 4.

```

"A igreja Xpto é constituída por um corpo principal e 2 torres.
O edifício principal está localizado em 0,0,0 e tem como
dimensões 6x15x3x1 e tem um telhado em forma de triângulo.
A torre A fica situada à direita e à frente e com dimensões
2x2x6x0.5 e um telhado em forma de prisma.
A torre sinaleira fica situada à esquerda e à frente e com
dimensões 2x2x6x0.5."
  
```

Figura 4: Extracto do esquema XML do modelo interno

Foram assim desenvolvidas várias gramáticas para extrair do texto as informações relevantes para o modelo interno, nomeadamente:

- O tipo de edifício (igreja, torre, catedral), século de construção e estilo arquitectónico;
- Constituição do edifício ou mais especificamente o número de torres;
- Localização do edifício principal;
- Dimensões das construções;
- Forma do telhado;
- Posição relativa das torres.

Para cada tipo de frase ou sequência foi criada uma gramática específica. A aplicação em cascata dessas gramáticas vai criar um ficheiro XML que contém as informações relevantes para este domínio e que representa o edifício na estrutura interna. O texto pode necessitar de alguma transformação, podendo-se utilizar XSLT.

```

<corpoPrincipal ID="main">
  <localizacao>
    <x>0</x>
    <y>0</y>
    <z>0</z>
  </localizacao>
  <features>
    <comp>4.0</comp>
    <larg>6.0</larg>
    <alt>6.0</alt>
    <vao>2.0</vao>
  </features>
</corpoPrincipal>
<torres>
  <torre ID="Torre A">
    <posicao>
      <posX desvio="-1.0">esquerda</posX>
      <posY desvio="1.0">frente</posY>
    </posicao>
    <features>
      <comp>1.0</comp>
      <larg>1.0</larg>
      <alt>9.0</alt>
      <vao>1.0</vao>
    </features>
  </torre>
</torres>

```

Figura 5: Igreja com uma torre, representada no modelo interno

3.2 Modelo interno

O modelo interno pode armazenar um ou vários edifícios, sendo que cada um é caracterizado por ser de um dado tipo (igreja, catedral ou outro a definir na ontologia), tendo uma determinada denominação (nome). É definida a origem do referencial (localização) para este edifício, que é constituído por um corpo principal (ou aba central), por uma ou várias torres e por uma ou várias entradas.

Todos os edifícios têm, obrigatoriamente, um corpo principal, onde se indica quais as suas características (comprimento, largura e altura), indicando também o tipo de inclinação do telhado, representado pelo seu vão. A figura 5 mostra parte do modelo interno de uma igreja, com uma torre colocada no lado esquerdo e completamente dentro do corpo principal.

A descrição de um edifício deve ser sempre efectuada com o observador virado para a entrada principal. Para cada torre para além das suas características (comprimento, altura, largura e vão) deve-se também indicar qual a sua posição relativamente ao edifício principal e aos eixos X e Y. Assim, a posX pode assumir o valor o valor “esquerda”, “centro” ou “direita”, sendo também indicado, em percentagem, quanto da torre fica (ou não) dentro do edifício principal. No caso da posY os valores possíveis que pode assumir são “frente”, “meio” ou “tras”. No caso de não ser indicado no texto qual é o valor dessa percentagem, o módulo de processamento de linguagem natural assume que a torre se encontra encostada ao edifício. A forma dos telhados também é associada de forma automática ao tipo de edifício, tendo a forma de pirâmide no caso das torres e de triângulo nos outros.

3.3 Criação do modelo 3D

O modelo interno contém informação semântica que identifica o tipo de edifício (corpoPrincipal ou torre) e também informação sobre qual a posição absoluta (no caso do corpo principal) ou posição relativa (no caso dos outros edifícios), e também informação quantitativa relativa às dimensões de cada edifício. A cada tipo de edifício corresponde uma forma geométrica simples, tal como foi definida para esta fase do protótipo. Desta forma, é possível converter este modelo interno para uma representação 3D. Como neste momento o modelo só contém informação relativa à forma principal de cada edifício, o modelo será muito limitado. Como o formato cityGML conjuga informação semântica com informação geométrica, foi este o formato escolhido.

A figura 6 mostra o processo de criação do modelo 3D para cada tipo de edifício. A partir do modelo interno, e para cada nó que representa uma estrutura para a qual está definida uma forma padrão, é criado um nó cityGML que pode ser visualizado numa aplicação.

Os edifícios são constituídos por duas componentes, uma base e um telhado ou topo formando um único edifício (*building*).

A criação do ficheiro cityGML pode ser feita de forma manual ou utilizando uma biblioteca gratuita disponibilizada pelo cityGML e denominada citygml4j, que permite

a criação de modelos 3D no formato cityGML utilizando a linguagem Java.

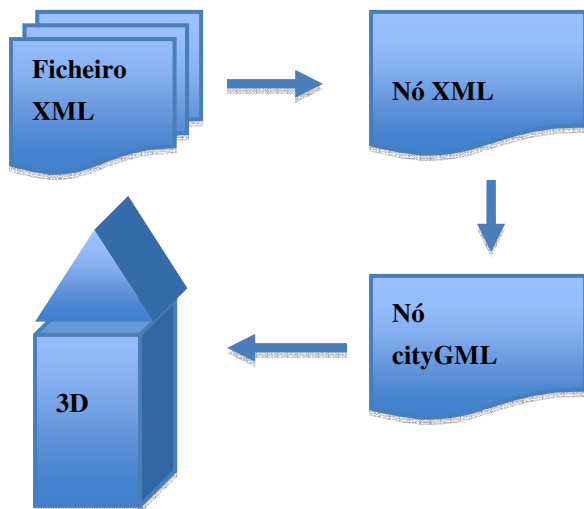


Figura 6: Processo de conversão para um modelo 3D

Foi desenvolvido um módulo que reconhece o tipo de edifícios incluídos no modelo interno (igreja, corpo principal, torres e entradas), processando-os um a um e criando o modelo 3D para cada elemento. A figura 7 representa a arquitectura do módulo que realiza esta tarefa.

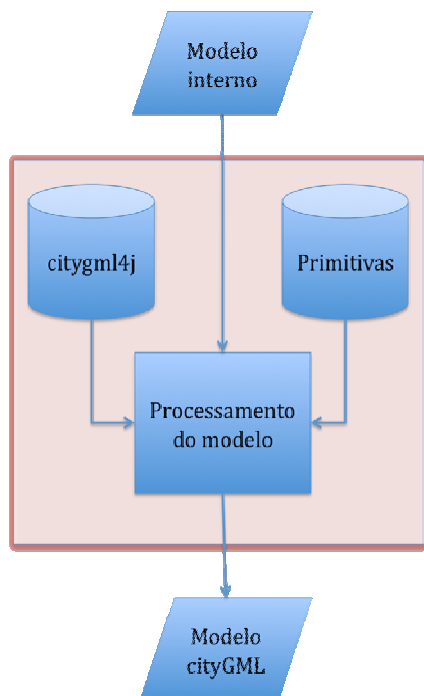


Figura 7: Conversão para o modelo cityGML

O modelo final é assim a junção destes vários modelos, sendo que cada edifício é sempre constituído por 3 componentes, a base (*ground*), as paredes (*walls*) e o telhado (*roof*), variando as suas características de acordo com o tipo de edifício. Nesta fase, para simplificação dos procedimentos, considerou-se que os edifícios tinham a forma base de um paralelepípedo rectângulo. A representação

destes componentes em cityGML é executada através de 3 primitivas, criadas previamente, e que permitem criar o modelo 3D em cityGML destes três tipos de componentes. Cada componente é assim transformada numa estrutura interna do cityGML que semanticamente corresponde a um *building*. A figura 8 mostra um modelo 3D representado no cityGML e a figura 9 mostra a constituição em blocos, em que cada bloco representa um componente.

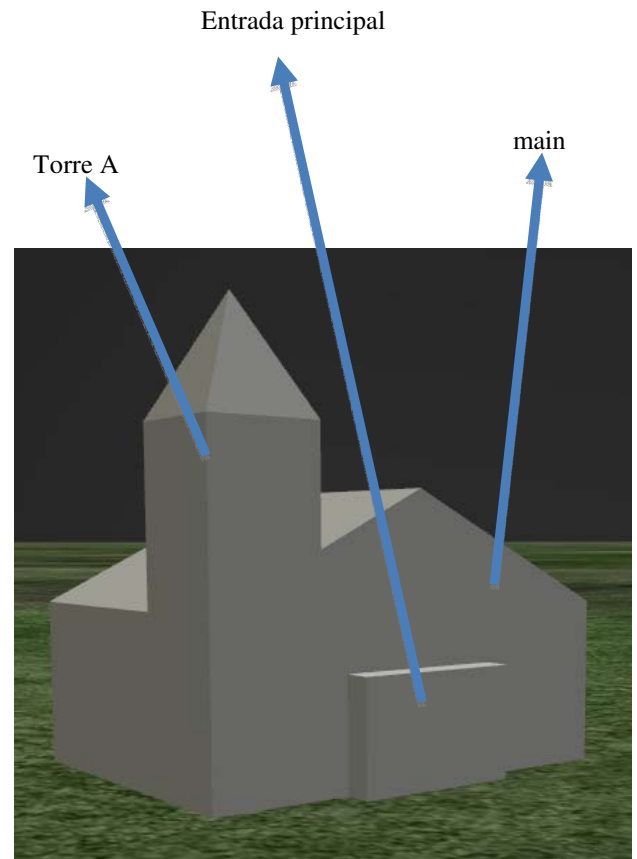


Figura 8: Modelo interno representado no cityGML

As funções que realizam essas operações foram denominadas:

- `ProcessMainBuilding(childNodes.item(loopIndex), citygml, gml, cityModel, origin);`
- `ProcessTowers(childNodes.item(loopIndex), citygml, gml, cityModel, origin, envelope);`
- `ProcessEntrances(childNodes.item(loopIndex), citygml, gml, cityModel, origin, envelope);`

Estas funções recebem vários parâmetros, dos quais se destacam o nó que está a ser processado, representado por `childNodes.item(loopIndex)` e o modelo cityGML que está a ser criado e que é armazenado na variável *citygml*. Todos eles recebem a origem do seu referencial, armazenado na variável *origin*, sendo que o processamento das torres e das entradas também recebe a informação sobre quais os limites do edifício principal, armazenado na variável *envelope*.

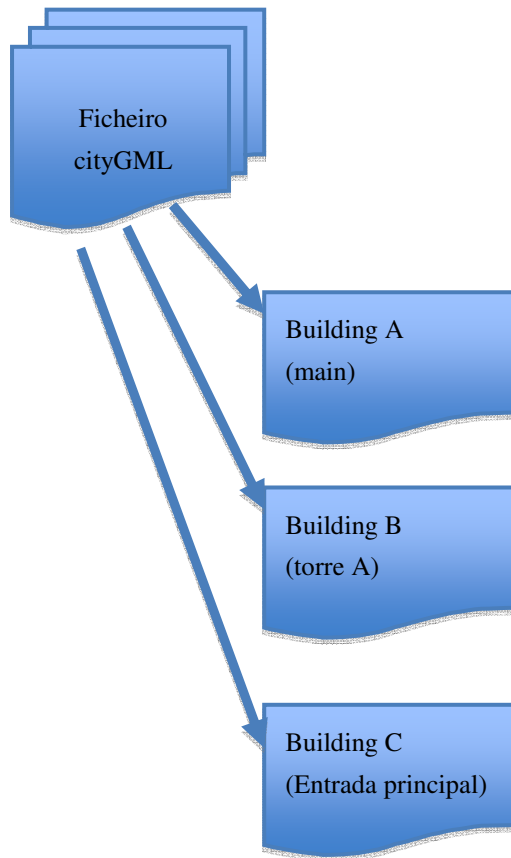


Figura 9: Representação do ficheiro de cityGML

4. RESULTADOS

O protótipo desenvolvido foi testado em duas fases distintas. Na primeira fase, foi testado o funcionamento do modelo tendo sido criados ficheiros XML que representassem alguns tipos de igrejas, variando as suas dimensões e o número e posicionamento das torres. A figura 10 mostra alguns dos resultados desses testes, neste caso representando múltiplos edifícios no ficheiro XML;



Figure 10: Representação do modelo interno

Na segunda fase foi testado o funcionamento do protótipo, utilizando alguns textos em linguagem natural. Foram utilizados textos no formato predefinido da gramática e outros em formato de texto livre.

4.1 Modelos 3D

A figura 11 mostra o texto e o modelo 3D de uma igreja com uma torre.

*“O edifício principal está localizado em 0,0,0 e tem como dimensões 4x6x6x2 e tem um telhado em forma de triângulo.
A torre principal fica situada à direita e à frente com as dimensões 2x2x9x2.
A entrada é feita pela porta situada ao centro e à frente com as dimensões 2x0,5x3x1 e um telhado em forma de triângulo.”*

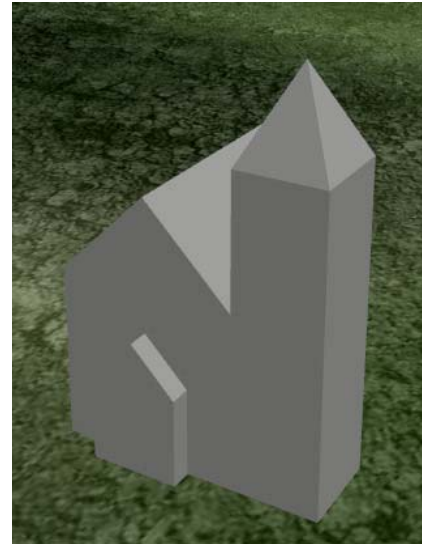


Figura 11: Representação de uma igreja com uma torre

Na conversão deste texto podemos observar que foram reconhecidas as 3 componentes (edifício principal, torre e entrada) tendo sido lidas correctamente as suas dimensões, mesmo sendo constituídas por números inteiros e números reais. A localização relativa da torre e da entrada também foi compreendida correctamente pelo módulo de processamento de linguagem natural. Relativamente à posição da torre, no texto não existe qualquer indicação sobre se esta está encostada ou embutida no edifício principal. O módulo de processamento de linguagem natural reconheceu na mesma o tipo de estrutura pretendida e ao criar o modelo interno adicionou a informação de que as torres por defeito se encontram encostadas ao edifício principal.

Na figura 12 está representada uma igreja com duas torres. Neste caso, tal como no texto da figura 11 foram reconhecidas as estruturas que se encontravam no texto e embora a descrição da capela seja semelhante à descrição da torre, o módulo de processamento de linguagem natural não o considerou pois não faz parte do Domínio definido. Também no caso das torres não foi necessário indicar qual era a forma do telhado, pois quando essa informação for omissa serão utilizados os valores por defeito, que no caso dos telhados das torres implica a forma em pirâmide.

“A igreja do Bonfim encontra-se em 0,0,0 e tem como dimensões 8x6x6x2 e tem um telhado em forma de triângulo.

Existe uma capela adjacente ao edifício no lado direito e à frente com as dimensões 2x4x4x1.

A torre A situa-se à direita e à frente embutida no edifício e com as dimensões 2x2x9x2.

A torre B fica situada à esquerda e à frente embutida com as dimensões 2x2x9x2.

A entrada é feita pela porta situada ao centro e à frente com as dimensões 2x0,5x3x1.”

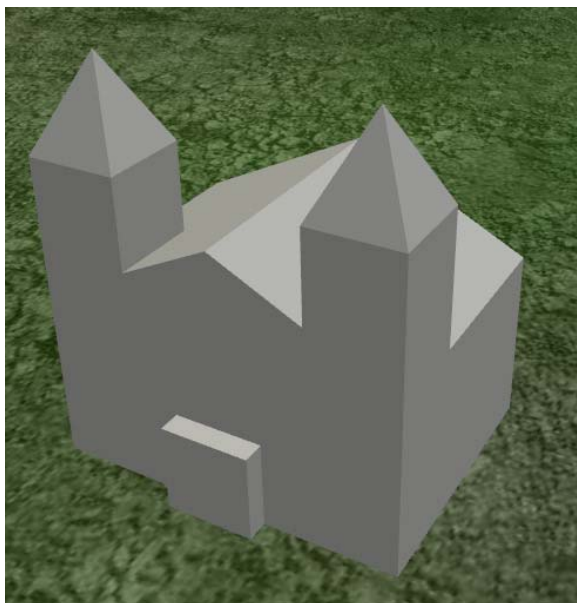


Figura 12: Representação de uma igreja com duas torres

Foram também criados modelos mais complexos, com mais torres como o exemplo da figura 13 com 4 torres ou modelos mais simples com uma única torre colocada no centro da face principal.

Embora não sejam situações típicas do formato de uma igreja, também foram testadas situações em que as torres não se encontram alinhadas com a face principal, como é mostrado na figura 10, no modelo mais à direita.

5. CONCLUSÃO E TRABALHO FUTURO

Os resultados demonstram que a estrutura interna do protótipo é suficientemente flexível para construir inúmeros modelos de igrejas, com um nível de realismo aceitável. Os resultados demonstram ainda que é possível criar um modelo 3D a partir de um texto escrito em português.

No caso do cityGML foi utilizado o nível de detalhe 2 sendo necessário aumentar para o nível de detalhe 3 para acrescentar mais pormenor e aumentar a fidelidade visual.

Para a criação de modelos 3D de igrejas tradicionais, cuja forma dos edifícios pode ser construída baseando-se numa paralelepípedo rectângulo, este modelo de modelação funciona pois todos os novos modelos podem ser construídos a partir da mesma base comum. Para outro

tipo de edifícios seria necessário encontrar uma solução alternativa.

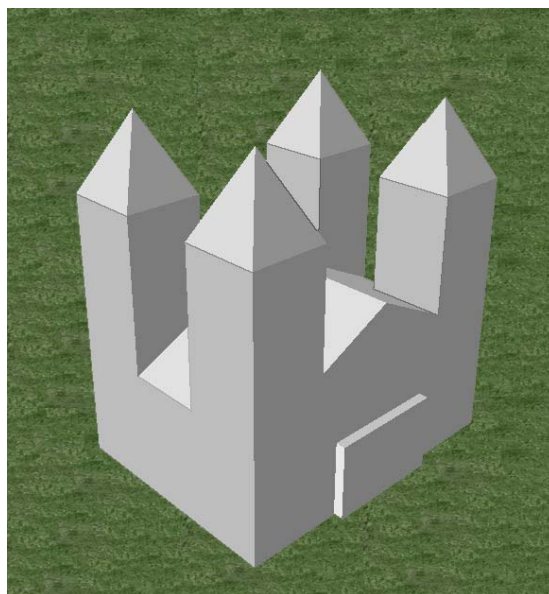


Figura 13: Igreja com 4 torres

Assim o trabalho futuro irá centrar-se em:

- permitir que possam ser definidas novas estruturas, baseadas nas formas primitivas e definidas segundo uma ontologia específica, de forma a expandir o tipo de modelos que possam ser utilizados;
- incluir a interactividade com o utilizador para quando for utilizada uma estrutura não reconhecida pela ontologia. Neste caso, de uma forma interactiva, o utilizador vai criando o novo modelo que passará a fazer parte da ontologia;
- partilhar a ontologia com o módulo de processamento de linguagem natural;
- aumentar o nível de fidelidade visual, adicionando o nível 3 aos modelos cityGML. Neste nível já serão representadas as fachadas das igrejas, aumentando o seu nível de detalhe;
- Criar modelos pré-definidos, eliminando a necessidade da introdução de valores quantitativos nas dimensões dos edifícios.

O modelo interno, resultante do processamento de linguagem natural, apresenta ainda uma vantagem. Para este protótipo foi utilizado para formato do modelo 3D o cityGML, mas poderia ter sido utilizado qualquer outro formato. Utilizando um modelo interno semântico associado a uma ontologia permite que sejam criados modelos 3D para conceitos, podendo ser utilizadas outras tecnologias, por exemplo as gramáticas de cisão para a construção dos modelos.

6. REFERÊNCIAS

- [Anslow06] - Anslow C.; Marshall, S.; Noble, J.; Biddle, R. Evaluating X3D for use in software visualization. *Proceedings of the 2006 ACM symposium on Software visualization*. (Brighton, United Kingdom 2006), 161-162.
- [Baeza-Yates04] - Baeza-Yates R. Challenges in the Interaction of Information Retrieval and Natural Language Processing. *Proc. 5th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing 2004)*. (2004). 445-456.
- [Bernardes03] - Bernardes P.; Martins, M. Computação Gráfica e Arqueologia Urbana: O caso de Bracara Augusta. *Actas do 12º Encontro Português de Computação Gráfica*. (2003). 63-72.
- [Berners-Lee01] - Berners-Lee T.; Hendler, J.; Lassila, O. The semantic web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American*. (2001),
- [Bontcheva02] - Bontcheva K.; Cunningham, H.; Tablan, V.; Maynard, D.; Hamza, O. Using GATE as an environment for teaching NLP. *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics*. (Philadelphia, Pennsylvania 2002), 54-62.
- [Coelho07] - Coelho A.; Bessa, M.; Sousa, A.; Ferreira, F. Expedient Modelling of Virtual Urban Environments with Geospatial L-systems. *Computer Graphics Forum*. 26:4 (2007),769-782.
- [Danahy95] - Danahy J.; Hoinkes, R. Polytrim: Collaborative Setting for Environmental Design. *CAAD Futures 95, The Global Design Studio*. (1995).
- [Döllner06] - Döllner J.; Kolbe, T.H.; Liecke, F.; Sgouros, T.; Teichmann, K. The Virtual 3D City Model of Berlin - Managing, Integrating, and Communicating Complex Urban Information. *Proceedings of the 25th Urban Data Management Symposium UDMS*. (2006).
- [Frischer08] - Frischer B. The Rome Reborn Project. How Technology is helping us to study history. (OpEd, November 10. University of Virginia 2008),
- [Garrette09] - Garrette D.; Klein, E. An Extensible Toolkit for Computational Semantics. *Proceedings of the Eighth International Conference on Computational Semantics*. (2009).
- [Guimarães02] - Guimarães F. Utilização de Ontologias no domínio B2C. *PUC-RJ*. (2002).
- [Kolbe05] - Kolbe T.H.; Gröger, G.; Plümer, L. CityGML - Interoperable Access to 3D City Models. *International Symposium on Geoinformation for Disaster Management*. (2005).
- [Liu06] - Liu Y.; Xu, C.; Pan, Z.; Pan, Y. Semantic modeling for ancient architecture of digital heritage *Computers & Graphics*. 30:5 (2006),800-814.
- [Mueller06] - Mueller P.; Wonka, P.; Haegler, S.; Ulmer, A.; Gool, L. Procedural Modeling of Buildings. *Proceedings of ACM SIGGRAPH*. (2006).
- [Mueller07] - Mueller P.; Zeng, G.; Wonka, P.; Gool, L. Image-based Procedural Modeling of Facades. *Proceedings of ACM SIGGRAPH*. (2007).
- [Pimentel01] - Pimentel J.; Baptista, N.; Goes, L.; Dionísio, J. Construção e gestão da complexidade de cenários urbanos 3D em ambientes virtuais imersivos. *Actas do 10º Encontro Português de Computação Gráfica*. (2001). 165-174.
- [Pollefeys00] - Pollefeys M.; Proesmans, M.; Koch, R.; Vergauwen, M.; Gool, L. Acquisition of Detailed Models for Virtual Reality. *British Archaeological Reports, International Series #843*. (2000).
- [Ribeiro08] - Ribeiro A. Port4NooJ: an open source, ontology-driven Portuguese linguistic system with applications in machine translation. *Proceedings of the 2008 International NooJ Conference (NooJ08)*. (2008).
- [Shiode01] - Shiode N. 3D urban models: Recent developments in the digital modelling of urban environments in three-dimensions. *GeoJournal*. 52 (2001),263-269.
- [Silberstein04] - Silberstein M. NooJ: A Cooperative, Object-Oriented Architecture for NLP. INTEX pour la Linguistique et le traitement automatique des langues. *Cahiers de la MSH Ledoux*. (2004).
- [Uotila00] - Uotila K.; Sartes, M. Medieval Turku – The Lost City. A Project trying to reconstruct a Medieval Town in Finland. *British Archaeological Reports, International Series #843*. (2000).
- [Wonka03] - Wonka P.W., M.; Sillion, F.; Ribarsky, W. Instant Architecture. *ACM Trans. Graph*. 22:3 (2003),669-677.