

Modelação Expedita de Cenas Urbanas

António Fernando Coelho
UTAD
Vila Real
acoelho@utad.pt

António Augusto de Sousa
FEUP / INESC Porto
Porto
aas@fe.up.pt

Fernando Nunes Ferreira
FEUP
Porto
fnf@fe.up.pt

Sumário

A natureza complexa e extensa dos ambientes urbanos torna morosa e dispendiosa a tarefa de construção de modelos virtuais destes. No entanto, a grande quantidade de informação disponível em formato digital sobre estas áreas motiva à automatização dos processos de modelação de forma a reduzir a intervenção humana.

Neste artigo apresenta-se um sistema para a modelação de ambientes urbanos baseado em informação disponível em formatos e plataformas diversas. A integração dos dados é efectuada utilizando XML, um standard para interoperabilidade. Constitui-se os processos de modelação, baseados na utilização de sistemas L e especificados num formato baseado em XML, denominado XL3D. O sistema permite a geração de forma automática de um modelo inicial, possibilitando o seu melhoramento de forma incremental através da adição de informação e/ou do refinamento dos processos de modelação.

Palavras-chave

Modelação 3D, Sistemas L, Realidade Virtual, XML, X3D, Computação Gráfica, Sistemas de Informação Geográfica

1. INTRODUÇÃO

A geração de modelos tridimensionais de áreas urbanas coloca um grande número de problemas no âmbito da Computação Gráfica, dado que a quantidade de informação necessária para a criação de modelos realistas é dependente da dimensão e complexidade da área a modelar. Embora existam alguns casos em que os modelos 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] [Dodge98].

1.1 Motivação

A natureza complexa dos ambientes urbanos, e a sua grande relevância socio-económica, levam inúmeros profissionais a realizar estudos sobre eles, recolhendo, analisando, processando e produzindo grande quantidade de informação. Actualmente, a grande maioria desta informação é armazenada em formato digital, na forma de ficheiros diversos ou em bases de dados.

Muita dessa informação disponível sobre áreas urbanas constitui um enorme potencial para servir de base à automatização do processo de criação de modelos tridimensionais de áreas urbanas através de processos automáticos. No entanto, a natureza abstracta, contextualizada e segmentada desta informação, bem como a diversidade de formatos e de plataformas em que se encontra disponível, dificultam o processo de desenvolvimento deste tipo de aplicações.

Verifica-se assim que existe um problema que é o de possibilitar a utilização de informação em formatos e plataformas diferentes. Uma solução para colmatar este problema passa pela utilização de standards de interoperabilidade como o XML (*Extended Modelling Language*) [XML]. Por outro lado, dado o carácter contextualizado e segmentado da informação relacionada com áreas urbanas, pode tornar-se necessária a integração de diversos documentos, bem como a dedução de alguma da informação que não está presente na informação de base.

Como exemplo, referira-se o caso da modelação de um edifício com base na informação disponibilizada num SIG (Sistema de Informação Geográfica). A informação geográfica relativa ao edifício é bidimensional e apenas representa a projecção desse edifício no elipsóide terrestre. Para se poder construir o poliedro correspondente, é necessária, no mínimo, alguma informação relacionada com a altura. Esta informação pode ser obtida de dados diversos, tais como o ponto de cénica (altitude do ponto mais elevado do edifício) ou o número de andares do edifício. Dado que esta informação se pode encontrar como atributo em temas geográficos distintos, ou numa base de dados relacional, torna-se necessária a sua integração. Adicionalmente, os edifícios podem ter estilos diferentes consoante a data de construção, sendo que esta pode ser encontrada na escritura e a sua arquitectura pode estar ligada à região da cidade em que se localiza.

A inadequação, ou mesmo falta de informação constitui assim um outro problema, cuja solução pode passar pela

capacidade de ampliação de dados, bem como pela introdução de alguma aleatoriedade, em partes do modelo em que a informação é escassa e não é necessário um elevado nível de realismo. Estas características são típicas de ferramentas matemáticas como os sistemas L [Prusinkiewicz90], que têm sido bastante utilizadas na Computação Gráfica, nomeadamente para a modelação de organismos e fenómenos naturais [Deussen98] [Prusinkiewicz01] [Lane02], embora existam poucos exemplos da sua utilização para a modelação de ambientes urbanos [Parish01].

A motivação principal deste trabalho é o desenvolvimento de um sistema de modelação para a criação de modelos virtuais de cidades reais. O sistema deve operar de forma semi-automática, utilizando informação existente previamente ou a recolher, para o processo de modelação. Dado o número elevado de elementos a modelar para cada cidade virtual e a variabilidade da quantidade de informação disponível em cada caso, o sistema deve ser capaz de deduzir informação, de forma a preencher lacunas na informação de base. A escassez de informação de modelação não deve, no entanto, ser impeditiva da obtenção de um elevado grau de detalhe do modelo virtual. Todo o processo de modelação deve ser controlado de forma concisa e genérica pelo utilizador, de forma a obter-se uma solução inicial, cujo realismo possa ser incrementalmente aumentado, através do refinamento da especificação do processo de modelação.

1.2 A Questão da Interoperabilidade

Ao nível da Computação Gráfica 3D, a questão da interoperabilidade tem sido abordada pelo consórcio Web3D [Web3D] através do desenvolvimento do X3D (*Extended 3D*) [X3D]. Sendo um desenvolvimento do VRML (*Virtual Reality Modelling Language*) [VRML97], o X3D possibilita a codificação num formato baseado em XML, sendo este um standard de interoperabilidade *de facto* para a representação e transferência de informação.

A informação geográfica é possivelmente o tipo de informação com maior relevância para a modelação dos espaços urbanos, uma vez que representa, em forma de mapas (representação bidimensional), aspectos geométricos e de localização geográfica, que são extremamente úteis para a geração dos modelos tridimensionais. Ao nível da interoperabilidade, também o consórcio *OpenGIS* [OpenGIS] tem desenvolvido alguns standards baseados em XML. A título de exemplo, refira-se o GML (*Geography Markup Language*) [GML] para representação da informação geográfica e o WFS (*Web Feature Service*) [WFS], uma especificação de serviços Web para disponibilização de informação geográfica. A integração que se propõem deste tipo de informação no processo de modelação é assim fundamental, dado que se pretendem desenvolver soluções automáticas.

As questões da reutilização dos processos de modelação, bem como da modularidade, são relevantes no aspecto

da eficiência e celeridade de criação de uma solução inicial por parte do utilizador. O projecto CONTIGRA [Dachsel02] é um claro exemplo de como a codificação XML do X3D pode contribuir claramente para a construção de modeladores que possibilitem a integração e reutilização de componentes no processo de construção de aplicações tridimensionais.

2. OS SISTEMAS L

Os sistemas L (acrónimo de Sistemas Lyndenmayer) foram introduzidos em 1968 por Astrid Lyndenmayer [Lyndenmayer68], e têm sido utilizados para a modelação da morfologia de uma grande variedade de organismos e entidades. A teoria matemática que os sustenta tem pontos de contacto com as teorias dos autómatos abstractos e das gramáticas formais.

Alguns dos pontos fortes dos sistemas L são a ampliação de dados [Smith84], que permite a geração de estruturas complexas a partir de um conjunto pequeno de dados, e a emersão. O conceito de emersão reflecte processos em que um conjunto de unidades interactuantes adquire, de forma qualitativa, novas propriedades que não podem ser deduzidas através de uma simples sobreposição das contribuições individuais [Taylor92]. Assim, estas novas propriedades “emergem” a partir da representação inicial.

Um dos conceitos fundamentais no contexto dos sistemas L, é o de módulo, designando uma unidade construtiva discreta que é repetida à medida que o sistema se desenvolve. Exemplo concreto deste conceito é o dos ramos e das folhas no caso da modelação de uma árvore.

A essência do desenvolvimento ao nível modular é capturada através de um sistema de reescrita paralela, que substitui individualmente os módulos ascendentes, por configurações de módulos descendentes.

Todos os módulos pertencem a um alfabeto finito de tipos de módulos e todo o comportamento de desenvolvimento modular é especificado através de um conjunto, também finito, de regras de reescrita ou de produção.

No seu caso mais simples, os sistemas L de contexto livre, cada regra de produção substitui um módulo predecessor, por zero, um, ou mais módulos sucessores. No caso dos sistemas sensíveis ao contexto, a aplicabilidade de uma determinada regra de produção não depende apenas do módulo predecessor, mas também dos seus vizinhos.

As regras de produção são aplicadas em paralelo, sendo reescritos todos os módulos simultaneamente em cada passo de derivação. A sequência de estruturas obtida em passos de derivação consecutivos a partir de uma estrutura inicial predefinida (axioma) denomina-se sequência de desenvolvimento.

Os sistemas L paramétricos [Prusinkiewicz95] estendem o conceito base dos sistemas L, com atributos numéricos em cada símbolo. Os sistemas L paramétricos operam com palavras paramétricas, ou seja, cadeias de módulos com parâmetros associados. Neste tipo de sistema, a

aplicação de uma regra de produção pode ser dependente da avaliação de uma condição.

Os sistemas L estocásticos [Prusinkiewicz90] adicionam o carácter da aleatoriedade, através da atribuição de um valor de probabilidade às regras de produção.

A nossa abordagem aos processos de modelação passa pela utilização de sistemas L sensíveis ao contexto, paramétricos e estocásticos.

3. O SISTEMA DE MODELAÇÃO XL3D

No presente documento apresenta-se um sistema de modelação para cenas urbanas que responde a uma especificação de modelação, gerando de forma automática um modelo tridimensional em formato X3D.

O sistema baseia-se numa abordagem procedimental, onde toda a modelação é especificada de forma declarativa, baseada em documentos segundo um XML-Schema [XML-Schema] denominado XL3D.

Os dados para modelação, em formatos baseados em XML, são convertidos, através de transformações XSLT [XSLT], em cadeias de módulos. Estas cadeias são concatenadas para cada um dos processos de modelação.

O processo de modelação é definido e controlado por sistemas L. Através de um conjunto de regras de produção, definidas pelo utilizador, os dados iniciais são transformados, através de um processo interactivo, até se obter uma solução satisfatória.

A solução final obtida é posteriormente interpretada, formando um *scenegraph* em formato X3D.

3.1 Processo de Modelação

Os sistemas L paramétricos conferem ao processo de modelação um elevado poder de ampliação de dados, assim como de emersão, gerando novos dados a partir do conhecimento associado às regras de produção definidas pelo utilizador. Por outro lado, a utilização de sistemas L estocásticos permite a incorporação de aleatoriedade no processo de modelação, de forma a simular a natureza diversificada dos cenários urbanos.

A primeira fase do processo de modelação consiste na formação de um axioma inicial, seguindo-se a sua transformação através da aplicação de regras de produção e, finalmente, a interpretação da cadeia resultante, para geração de um segmento de *scenegraph* em formato X3D.

3.1.1 Axioma

Em primeiro lugar torna-se necessário criar uma cadeia inicial de módulos que sirva como axioma.

A cadeia inicial é construída a partir de diversos ficheiros XML que são transformados em cadeias de módulos através de transformações XSLT. O resultado final da transformação deve obedecer ao XML-schema apresentado na figura 1.

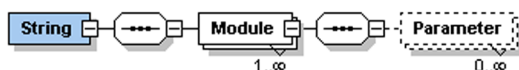


Figura 1 – Esquema da cadeia de módulos

```
<wfs:Edificado fid="Edificado.1">
  <wfs:NPisos>5</wfs:NPisos>
  <wfs:Altitude>435</wfs:Altitude>
  <wfs:Geometry2>
    <gml:Polygon srsName="Port:ugal">
      <gml:outerBoundaryIs>
        <gml:LinearRing>
          <gml:coordinates>
            524000.084,      211684.916
            524004.656,      211684.519
            524005.727,      211695.958
            524007.280,      211696.026
            524007.608,      211703.813
            524007.605,      211703.813
            524000.084,      211704.467
            524000.084,      211684.916
          </gml:coordinates>
        </gml:LinearRing>
      </gml:outerBoundaryIs>
    </gml:Polygon>
  </wfs:Geometry2>
</wfs:Edificado>
```

Figura 2 - Segmento de ficheiro GML

Por exemplo, partindo de um elemento de edificado representado num ficheiro GML (figura 2) e após aplicação de uma transformação XSLT, obtém-se uma cadeia de módulos como se mostra na figura 3.

```
<XL3D:String>
  <XL3D:Module name="Origem">
    <XL3D:parameter>
      524000.08
    </XL3D:parameter>
    <XL3D:parameter>435</XL3D:parameter>
    <XL3D:parameter>
      2116849.16
    </XL3D:parameter>
  </XL3D:Module>
  <XL3D:Module name="Edificio">
    <XL3D:parameter>5</XL3D:parameter>
    <XL3D:parameter>
      0.000 0.000, 4.572 -0.397,
      5.643 11.042, 7.196 11.110,
      7.524 18.897, 7.521 18.897,
      0.000 19.551, 0.000 0.000
    </ XL3D:parameter>
  </XL3D:Module>
</XL3D:String>
```

Figura 3 - Cadeia de módulos

Como se pode ver, a cadeia resultante é constituída por módulos paramétricos, compostos por um nome e por uma sequência de parâmetros. Na criação de um axioma pode utilizar-se uma destas cadeias, ou integrar várias, segundo um processo de concatenação ou de composição através de uma transformação XSLT.

3.1.2 Transformação

A transformação do axioma (figura 4) é realizada através de um conjunto de regras de produção definidas pelo utilizador. Nesta fase, o conhecimento do utilizador sobre

o problema é utilizado para ampliar o conjunto de dados associados ao axioma.

```
Origem(524000.084, 435, 211684.916)
Edificio(5, "0.000 0.000, 4.572 -0.397,
5.643 11.042, 7.196 11.110, 7.524
18.897, 7.521 18.897, 0.000 19.551
0.000 0.000")
```

Figura 4 - Axioma

Cada regra de produção é aplicada a uma determinada cadeia de módulos parametrizados.

A notação utilizada para a especificação das regras de produção é a dada por:

$$n: E < P > D : cond \xrightarrow{prob} S_1 \dots S_n \quad (1)$$

Onde, *P* representa o módulo a identificar e a substituir por um ou vários módulos sucessores *S*; *E* representa os módulos vizinhos situados à esquerda; *D* os módulos vizinhos situados à direita, *cond* é a condição da qual depende a aplicabilidade da regra, *prob* a probabilidade da regra de produção e *n* o número da regra de produção.

Uma determinada regra de produção só é aplicada se existir um módulo *P* na cadeia a transformar e se a condição associada for verdadeira. No caso de ser definido um contexto, ou seja um conjunto de módulos situados na vizinhança (à sua esquerda ou à sua direita) do módulo predecessor, também é necessário que estes coincidam para que a regra seja aplicável.

Sempre que existam várias regras de produção aplicáveis para cada módulo predecessor a substituir, selecciona-se aleatoriamente uma delas, com base na probabilidade associada.

Só após efectuar a avaliação e selecção das regras de produção, é que estas se aplicam sincronamente a toda a cadeia, gerando a cadeia seguinte.

2:Edificio(npisos, contorno) $\xrightarrow{0.25}$ EdificioF1(npisos, contorno, 1)
3:Edificio(npisos, contorno) $\xrightarrow{0.25}$ EdificioF1(npisos, contorno, 2)
4:Edificio(npisos, contorno) $\xrightarrow{0.25}$ EdificioF1(npisos, contorno, 3)
5:Edificio(npisos, contorno) $\xrightarrow{0.25}$ EdificioF1(npisos, contorno, 4)
6:EdificioF1(npisos, cont, mat) $\xrightarrow{0.5}$ EdificioF2(npisos, cont, mat, 1)
7:EdificioF1(npisos, cont, mat) $\xrightarrow{0.5}$ EdificioF2(npisos, cont, mat, 2)

Figura 5 - Regras de produção iniciais

Este processo repete-se sucessivamente, até que se chegue a uma situação em que mais nenhuma regra de

produção seja aplicável, ou que se tenha atingido um número máximo de iterações.

A título de exemplo, o axioma apresentado na figura 4 pode ser transformado, de acordo com um conjunto de regras de produção, numa cadeia que descreve a estrutura de um edifício para a geração do *scenegraph* do seu modelo tridimensional em X3D.

Na figura 5 apresenta-se um subconjunto de regras de produção, do tipo estocástico, que permitem atribuir aleatoriamente um determinado material e tipo de telhado ao edifício a ser gerado (figura 6).

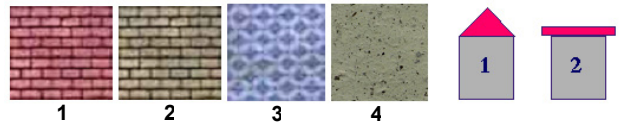


Figura 6 - Materiais de parede e tipo de telhado

Um outro conjunto de regras de produção expandem o contorno 2D inicial numa sequência de arestas, calculando o valor do seu perímetro, bem como criam um contorno tridimensional.

Na figura 7 apresenta-se a regra que constrói a estrutura base do *scenegraph* final. O edifício é transformado num LOD (*Level Of Detail*) com um modelo pormenorizado visível até 250 metros de distância, e um modelo mais simples, só com o bloco do edifício, para distâncias superiores.

```
22:Translacao(x,y,z)<EdificioF4
(npisos, cont, mat, telhado, perim, cont3
D, npts) -> LOD(0,0,0,100) [ Grupo
[ Predio(mat, cont, npisos*3, perim) ]
[ Translacao(0, npisos*3, 0)
[ Telhado(telhado, cont, perim) ]
[ Caleira(mat, cont3D, perim) ] ]
[ Rebordo(mat, cont3D, perim) ]
[ ConstFachada(mat, cont3D, npts,
npisos*3) ]
[ ConstRC(mat, cont3D, perim, npts) ]
] PredioSimples(mat, cont, npisos*3) ]
```

Figura 7 - Regra de produção que estrutura o edifício

O modelo pormenorizado é composto por um bloco (volumetria), telhado, caleira, rebordo da base, rés do chão e restante fachada.

```
31:
InstElemFachadaF4(npisos, mat, x1, y1, dx,
dy, nbandas, banda, ang) : banda > 0
-> InstJanelaF1(npisos, mat, x1+dx
*(2*banda-1)/(2*nbandas), y1+dy*
(2*banda-1)/(2*nbandas), ang)
InstElemFachadaF4(npisos, mat, x1, y1,
dx, dy, nbandas, banda-1, ang)
```

Figura 8 - Regra para a construção de bandas

O contorno do edifício é decomposto nas diferentes faces da fachada do edifício. Para cada face é calculada a sua

largura, e a partir deste valor, a regra 31 (figura 8) determina o número de bandas de janelas ou varandas.

Na figura 9 apresentam-se as regras de produção que instanciam os elementos da fachada que integram as bandas. As regras 33 e 34 determinam aleatoriamente se essa banda é uma janela ou varanda, sendo a probabilidade da primeira duas vezes superior à segunda. Na regra 35 decompõe-se cada banda nas posições relativas aos diferentes pisos do edifício. Para dar um aspecto mais realista ao edifício, nas regras 36 a 39, selecciona-se aleatoriamente se a janela deve estar com a persiana aberta ou entreaberta.

33: InstJanelaF1 (npisos, mat, x, y, ang) → ^{0.67} InstJanelaF2 (npisos, mat, x, y, ang, 0)
34: InstJanelaF1 (npisos, mat, x, y, ang) → ^{0.33} InstJanelaF2 (npisos, mat, x, y, ang, 1)
35: InstJanelaF2 (nps, mat, x, y, a, v) : npisos > 1 → [InstJan (x, 3*(nps-1)+1.5, y, a, mat, v)] InstJanelaF2 (nps-1, mat, x, y, a, v)
36: InstJan (x, y, z, a, m, v) :v=0 → ^{0.67} Jan (x, y, z, a, m)
37: InstJan (x, y, z, a, m, v) :v=1 → ^{0.67} Varan (x, y, z, a, m)
38: InstJan (x, y, z, a, m, v) :v=0 → ^{0.33} Jan2 (x, y, z, a, m)
39: InstJan (x, y, z, a, m, v) :v=1 → ^{0.33} Varan2 (x, y, z, a, m)

Figura 9 - Regras de produção para a fachada

Após a execução de todas as iterações, o sistema L termina a sequência de desenvolvimento e, para o caso exemplificado, obtém-se a cadeia de módulos representada na figura 10.

3.1.3 Interpretação

Depois de obtida uma representação de um determinado objecto para modelação, é necessário interpretar essa descrição de forma a gerar o segmento de *scenegraph* correspondente, em formato X3D.

Nesta fase o utilizador tem a liberdade de definir a forma como cada um dos módulos paramétricos é convertido para X3D.

Cada módulo paramétrico é substituído por um segmento do *scenegraph* em que determinados parâmetros deste gráfo são instanciados pelos argumentos do módulo. Assim, a cadeia de módulos final é interpretada como uma estrutura hierárquica em forma de árvore, em que o primeiro módulo (mais à esquerda) é a raiz, e o último módulo uma folha. De forma a permitir a criação de

“ramos” paralelos, existe um conjunto de módulos adicionais (representados pelos símbolos [e]) que os delimitam.

```
[Translação (x, y, z)
LOD (0, 0, 0, range)
Grupo
  [Predio (mat, cont, alt, perim) ]
  [Translacao (x, y, z)
    [Telhado (tipo, cont, perim) ]
    [Caleira (mat, cont3D, perim) ] ]
  [Rebordo (mat, cont3D, perimetro) ]
  [Jan (x, y, z, a, m) ] [Jan2 (x, y, z, a, m) ]
  [Jan (x, y, z, a, m) ] [Jan (x, y, z, a, m) ]
  [Varan (x, y, z, a, m) ] [Varan (x, y, z, a, m) ]
  [Varan (x, y, z, a, m) ] [Varan2 (x, y, z, a, m) ]
  [Jan2 (x, y, z, a, m) ] [Jan2 (x, y, z, a, m) ]
  [Jan2 (x, y, z, a, m) ] [Jan (x, y, z, a, m) ]
  [Jan2 (x, y, z, a, m) ] [Jan (x, y, z, a, m) ]
  [Jan2 (x, y, z, a, m) ] [Jan2 (x, y, z, a, m) ]
  [Jan2 (x, y, z, a, m) ] [Jan2 (x, y, z, a, m) ]
  [Varan2 (x, y, z, a, m) ] [Varan (x, y, z, a, m) ]
  [Varan2 (x, y, z, a, m) ] [Varan (x, y, z, a, m) ]
  [Varan (x, y, z, a, m) ] [Varan2 (x, y, z, a, m) ]
  [Jan (x, y, z, a, m) ] [Jan2 (x, y, z, a, m) ]
  [Jan (x, y, z, a, m) ] [Jan2 (x, y, z, a, m) ]
  [Jan (x, y, z, a, m) ] [Jan (x, y, z, a, m) ]
  [Jan2 (x, y, z, a, m) ] [Jan (x, y, z, a, m) ]
  [Jan (x, y, z, a, m) ] [Jan2 (x, y, z, a, m) ]
  [Jan (x, y, z, a, m) ] [Jan (x, y, z, a, m) ]
  [Porta (x, y, z, a, m) ] ]
PrédioSimples (mat, cont, altura) ]
```

Figura 10 - Cadeia final representando um edifício

Prusinkiewicz foi um dos pioneiros ao nível da utilização dos sistemas L na Computação Gráfica e, num dos seus trabalhos [Prusinkiewicz86], introduziu um método de interpretação baseado na metáfora da “tartaruga” utilizada na linguagem LOGO [Abelson82]. Nessa abordagem, um cursor 3D desloca-se num espaço tridimensional seguindo uma interpretação da cadeia final, através de um vocabulário restrito de símbolos que alteram o seu posicionamento e orientação. Ao longo do trajecto, vão sendo instanciadas primitivas tridimensionais.

A abordagem utilizada neste trabalho é diferente e baseia-se na interpretação de uma estrutura hierárquica em árvore representando o *scenegraph*, em que cada módulo é convertido, por meio de protótipos definidos pelo utilizador, para segmentos de *scenegraph* em X3D. Os protótipos baseiam-se no nó PROTO do X3D, sendo permitidas expressões para os campos.

Partindo da cadeia final que representa um edifício e após a sua interpretação por um conjunto de protótipos, obtém-se como representação final o *scenegraph* em X3D representado na figura 11.

3.1.4 Refinamento incremental do modelo

A solução obtida após a aplicação inicial de um determinado sistema L nem sempre produz os melhores

resultados ou um nível de realismo suficiente para a aplicação em causa.

Nestes casos, o utilizador pode ir refinando o conjunto de regras de produção no sentido de atingir a solução desejada, ou pode ainda aumentar a informação descrita no axioma, colectando maior quantidade de dados.

O sistema é assim de refinamento incremental: permite obter rapidamente um modelo grosseiro mas coerente de uma área urbana, sendo que o utilizador pode, à medida das suas necessidades, melhorá-lo globalmente ou em detalhes particulares.



Figura 11 – Modelo final do edifício em X3D

3.2 Especificação de Modelação (XL3D)

Todo o processo de modelação é especificado de forma declarativa através de um documento baseado num XML-Schema definido especificamente para este sistema, denominado XL3D [Coelho03].

A estrutura do documento é hierárquica e a raiz do documento é o projecto XL3D (*XL3Dproject*, figura 12). Cada documento contém um único projecto, sendo este constituído por um cabeçalho, modelos, procedimentos de modelação, protótipos e fontes de dados.

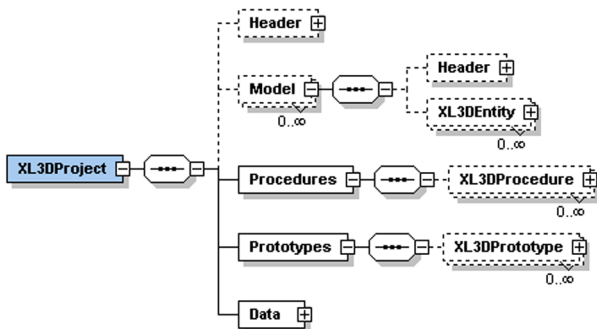


Figura 12 – Projecto XL3D

O cabeçalho (*Header*, figura 13) foi definido de forma a proporcionar informação de autoria, nomeadamente o seu título, o nome dos autores, a versão e alguma documentação. Devido à sua natureza estruturante, este elemento faz parte integrante da maioria dos restantes elementos.

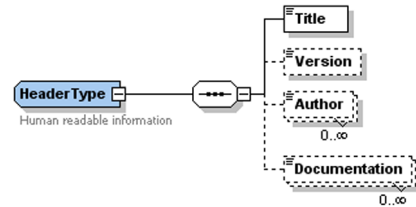


Figura 13 – Cabeçalho

Cada modelo (*Model*, figura 12) é composto por entidades, reflectindo a natureza complexa e estruturada dos ambientes urbanos, segundo uma visão subjectiva do utilizador. As entidades (*Entity*, figura 14) encapsulam um conjunto de modelos geométricos correspondentes às diversas entidades que compõem o projecto. Sendo definida numa estrutura hierárquica, uma entidade pode conter outras entidades cujo modelo pode ser instanciado em qualquer posição do *scenegraph*.

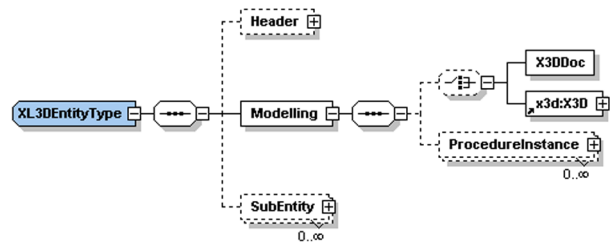


Figura 14 – Entidade

Como exemplo, um modelo de uma cidade pode ser composto por várias entidades como a rede viária, a vegetação e o edificado. A rede viária, por sua vez também poderá ser composta pelas entidades pavimento, passeios, sinalização, etc. Cada entidade instancia um conjunto de procedimentos de modelação que geram os modelos tridimensionais a partir de cadeias de módulos, obtidas de fontes de dados diversas.

O axioma é composto a partir de uma ou várias fontes de informação, sendo composto por concatenação ou aplicação de uma transformação XSLT.

Os procedimentos de modelação (*XL3DProcedure*, figura 15) constituem o componente nuclear do processo de modelação e baseiam-se em regras de produção que controlam o Sistema L que modela partes específicas de cada entidade.

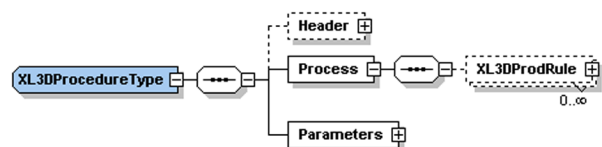


Figura 15 – Procedimento de modelação

Os procedimentos de modelação são definidos como componentes reutilizáveis que podem ser instanciados de forma a modelar distintas entidades e que podem ser estendidos para constituir novos procedimentos de modelação.

Cada procedimento de modelação possui assim um conjunto de regras de produção que transformam uma cadeia inicial de módulos, denominada axioma. As regras de produção (*XL3Dproduction*, figura 16) seguem a fórmula representada em (1) relativas a sistemas L sensíveis ao contexto, paramétricos e estocásticos.

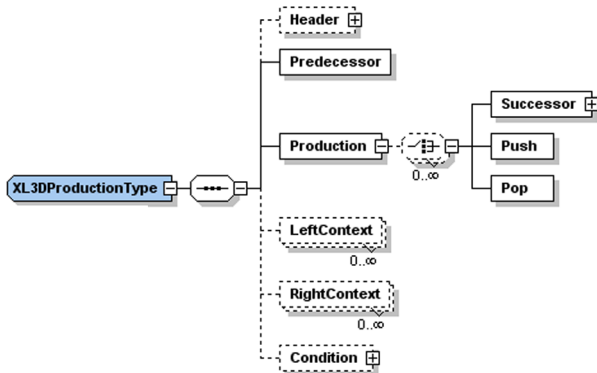


Figura 16 – Regra de produção

A cadeia de módulos resultante é interpretada de forma a criar um *scenegrph*, através da instanciação de um conjunto de protótipos.

Os protótipos (*Prototype*, figura 17) são segmentos de um grafo de cena X3D que podem ser instanciados através da interpretação da cadeia final de módulos obtida dos procedimentos de modelação.

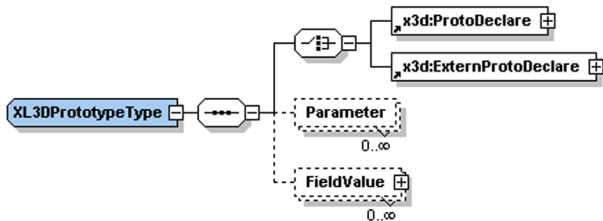


Figura 17 – Protótipo

As fontes de dados (*DataSources*, figura 18) contêm referências a dados provenientes de ficheiros XML. Os dados são convertidos do seu formato original para uma cadeia de módulos de informação através das transformações XSLT.

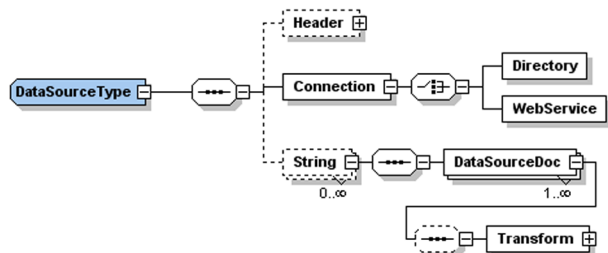


Figura 18 – Fonte de dados

4. RESULTADOS OBTIDOS

Ao longo da secção 3.1 exemplificou-se o processo de modelação de um elemento de edificado partindo de dois módulos. Após o processo de modelação, obteve-se uma cadeia final representando um edifício, composta por 46 módulos. Depois da sua interpretação foi gerado um ficheiro X3D com 277 nós. Demonstra-se assim o poder de ampliação de dados que este sistema possui, bem como a capacidade de emersão, ao gerar um objecto tão complexo como um edifício, partindo de algo tão simples como um contorno geométrico e o número de pisos. Existe efectivamente uma mais valia, quando comparamos com o processo de modelação manual, em que seria necessário posicionar, orientar e caracterizar cerca de 45 elementos distintos para a construção deste edifício.

O processo, que foi aplicado a um só elemento para exemplificação, pode ser aplicado, da mesma forma, a axiomas contendo grande quantidade de elementos, o que permite gerar cenas urbanas extensas sem qualquer esforço acrescido por parte do utilizador. No entanto, o processo de modelação manual obrigaria o utilizador a modelar individualmente cada um dos edifícios, aumentando o esforço envolvido de forma proporcional ao número de edifícios. Utilizando grupos de procedimentos de modelação e associando-os a vários axiomas representando informação diversificada, conseguem-se modelar os diferentes elementos que estruturam e compõem um ambiente urbano.

Os processos de modelação especificados para um determinado caso podem facilmente ser reutilizados para serem aplicados a outros com características semelhantes, estendendo o conjunto das suas regras de produção.

As figuras 19 e 20 exemplificam a potencialidade do sistema desenvolvido para a modelação de ambientes urbanos, para a geração de forma automática de soluções iniciais com grau de realismo aceitável.

O modelo foi gerado a partir da informação contida num sistema de informação geográfica, sendo a informação geográfica (figura 21), disponível à escala 1:2000, estruturada segundo 3 temas: espaços verdes, rede viária e edificado.

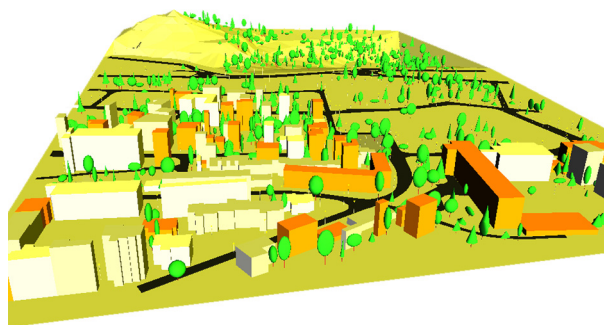


Figura 19 – Modelo virtual de ambiente urbano



Figura 20 – Vista de pormenor do modelo virtual



Figura 21 - Informação geográfica disponível

5. CONCLUSÕES E TRABALHO FUTURO

O sistema de modelação desenvolvido pode ser utilizado com grandes vantagens na geração de modelos tridimensionais de zonas urbanas reduzindo, quer os custos, quer a duração do processo.

O modelo obtido pode ser incrementalmente melhorado, através do refinamento das regras que controlam o processo de modelação, bem como através da adição de mais informação.

No caso específico de alguns elementos complexos do modelo, como é o caso de monumentos ou edifícios emblemáticos, o sistema não garante facilmente o nível de realismo desejado, pelo que estes devem ser modelados de forma cuidada em outro tipo de modeladores e posteriormente integrados no processo de modelação do sistema.

O sistema pode produzir resultados de elevado nível de detalhe e realismo, sempre que a informação disponível seja vasta e de qualidade. No entanto, mesmo nos casos em que tal informação é escassa, o sistema é capaz de chegar a uma solução inicial que, embora possua menor nível de realismo, mantém um nível de detalhe aceitável por via da incorporação de alguma aleatoriedade.

A definição do conjunto de regras de produção para a modelação de uma determinada entidade não é trivial, sendo um processo minucioso e iterativo. No entanto, a utilização de procedimentos de modelação paramétricos,

bem como a facilidade com que se consegue estender um determinado processo para modelar outra entidade, obviam essa dificuldade, possibilitando a reutilização de muitos dos processos de modelação.

Em termos de trabalho futuro, é essencial criar uma ferramenta de apoio à especificação do projecto de modelação, com especial ênfase na criação dos conjuntos de regras de produção que formam os diversos processos de modelação.

Pretende-se ainda implementar este sistema como um serviço Web, de forma a estar disponível para integração com outras aplicações, numa arquitectura distribuída.

Como exemplo da relevância desta arquitectura, está a ser desenvolvido um sistema móvel de navegação em tempo real [Pinto03] onde se pretendem integrar modelos tridimensionais, do trajecto e da envolvente. Dado que estes trajectos são determinados em tempo real em resposta a um pedido do utilizador, os modelos têm que ser gerados dinamicamente a partir daquele serviço Web.

Outro desenvolvimento possível é a integração de outros serviços Web, como fonte de dados para o processo de modelação. Dados os avanços na especificação de serviços Web para disponibilização de informação geográfica, como o WFS [WFS], o acesso ao resultado de interrogações avançadas à base de dados e de operações espaciais ficaria simplificado.

Também ao nível da interacção com o utilizador se pretende desenvolver um conjunto de ferramentas que permitam a geração dos ficheiros XL3D de forma intuitiva, bem como a integração no processo de informação raster proveniente de imagens captadas sobre o modelo real.

6. REFERÊNCIAS

- [Abelson82] Abelson H., diSessa A.: Turtle geometry, M.I.T. Press, Cambridge, 1982.
- [Coelho03] Coelho A., Sousa A., Ferreira F.: 3D Modelling of Large Urban Scenes from Diverse Sources of Information. *Proceedings of the 7th ICCO/IFIP International Conference on Electronic Publishing*, pp. 278-287, 2003.
- [Dachselt02] Dachselt R., Hinz M., Meißner K.: CONTIGRA: An XML-Based Architecture for Component-Oriented 3D Applications, In *Proceedings of Web3D '02*, pp. 155-163, 2002.
- [Deussen98] Deussen O., Hanrahan P., Lintermann B., Mech R.: Realistic modelling and rendering of plant ecosystems. *Proceedings of SIGGRAPH 98*, 1998.
- [Dodge98] Dodge M., Doyle S., Smith A., Fleetwood S.: Towards the Virtual City: VR & Internet GIS for Urban Planning. Paper presented at the *Virtual Reality and Geographical Information Systems Workshop*, London, 1998.
- [GML] GML - Geography Markup Language. <<http://www.opengis.net/gml/>>

- [Lane02] Lane B., Prusinkiewicz P.: Generating spatial distributions for multilevel models of plant communities. *Proceedings of Graphics Interface 2002*, pp. 69-80, 2002.
- [Lindenmayer68] Lindenmayer, A.: Mathematical models for cellular interaction in development, Parts I and II. *Journal of Theoretical Biology*, 18, pp. 280-315, 1968
- [OpenGIS] The OpenGIS Consortium, Inc.
<<http://www.opengis.org/>>
- [Parish01] Parish Y., Muller P.: Procedural Modeling of Cities. *Proceedings of SIGGRAPH 2001*, pp. 301-308, 2001.
- [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*, pp. 165-174, 2001.
- [Pinto03] Pinto P., Serôdio C., Coelho A., Mestre P., Soares S., Amaro C., Ferro P.: Audio Interface for Real Time Navigation. *Proceedings of the 4th Conference in Telecommunications*, pp. 337-341, 2003.
- [Prusinkiewicz01] Prusinkiewicz P., Muendermann L., Karwowski R., Lane B.: The use of positional information in the modeling of plants. *Proceedings of SIGGRAPH 2001*, pp. 289-300, 2001.
- [Prusinkiewicz86] Prusinkiewicz P.: Graphical applications of L-systems. In *Proceedings of Graphics Interface '86 – Vision Interface '86*, pp. 247-253, 1996.
- [Prusinkiewicz90] Prusinkiewicz P., Lindenmayer A.: *The Algorithmic Beauty of Plants*, Springer, 1990.
- [Prusinkiewicz95] Prusinkiewicz P., Hammel, M., Mech R., Hanan J.: *The Artificial Life of Plants. From Artificial life for graphics, animation and virtual reality*, volume 7 of SIGGRAPH '95 Course Notes, pages 1-1-1-38. ACM Press, 1995.
- [Smith84] Smith A.; Plants, fractals and formal languages. *Proceedings of SIGGRAPH '84 in Computer Graphics*, 18, 3, pp. 1-10, ACM SIGGRAPH, New York, 1984.
- [Taylor92] Taylor, C.: “Fleshing out” Artificial Life II. In C. G. Langton, C. Taylor, J. D. Farmer and S. Rasmussen, editors, *Artificial Life II*, pp. 25-38, Addison Wesley, Redwood City, 1992.
- [VRML97] The VRML Consortium Inc.: “The Virtual Reality Modelling Language – International Standard ISO/IEC 14772-1:1997”.
http://www.web3d.org/technicalinfo/specifications/ISO_IEC_14772-All/index.html
- [Web3D] The Web3D Consortium.
<http://www.web3d.org/>
- [WFS] Web Feature Service
<http://www.opengis.org/techno/RFC13.pdf>
- [X3D] Web3D Consortium: “X3D: The Virtual Reality Modelling Language – International Standard ISO/IEC 14772:200x”.
<http://www.web3d.org/TaskGroups/x3d/specification/>
- [XML] XML - Extensible markup Language.
<<http://www.w3.org/XML/>>
- [XML-Schema] XML-Schema.
<<http://www.w3.org/XML/Schema>>
- [XSLT] XSLT- Extended Stylesheet Language Transformations (XSLT).
<<http://www.w3.org/TR/xslt11/>>