Expeditious Modelling of Urban Scenes

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

Abstract

The complex and extensive nature of urban environments creates difficulties to the task of generating virtual models. However, the amount of information regarding urban areas available in digital format is an enabling condition to the automation of the modelling processes, thus reducing the amount of human intervention required.

This paper presents a three-dimensional modelling system based on interoperable access to data in diverse formats and digital support. Data integration is achieved by the use of XML documents or invoking Web services in a distributed architecture.

The system is driven by L-systems based modelling processes that automatically generate initial solutions for virtual environments. These initial solutions are, probably, characterised by a low level of realism, but can be incrementally improved.

The modelling processes and data sources are specified in a declarative way, using documents based on an XML-schema called XL3D.

Key-words

Computer Graphics, 3D Modelling, Virtual Reality, XML, X3D, L-Systems, Geographic Information Systems.

1. INTRODUCTION

Generating models of urban environments poses a great number of computer graphics problems, as the amount of information needed to create realistic models increases with the scene size and complexity. Although some models have been created by individual modelling of each element in the urban environment, semi-automatic methods [Pimentel01] [Dodge98] have proven to be much more effective by reducing the duration, the cost and the allocation of human resources.

1.1 Motivation

The complex nature of urban environments, and its socio-economics relevance, drives innumerable groups of professionals to carry through studies on these areas, thereby collecting, analysing, processing and producing large amounts of information. Currently, the great majority of this information is stored in files or databases and constitutes an invaluable asset to the generation of three-dimensional models of urban areas through automatic processes.

The diversity of digital formats and supporting platforms for the information poses a common problem, known as an interoperability problem. A solution to overcome this problem is the use of standards like XML (Extended Markup Language) [XML]. On the other hand, the integration of different types of documents, as well as the deduction of additional information that is not present in the data base, may be necessary, due to the segmented and contextualised nature of information regarding urban areas.

Considering the task of modelling a building from an urban area as an example: the available geographic information in a GIS (Geographic Information System) is usually just a two-dimensional projection of the building volume over the terrestrial ellipsoid. To be able to construct the corresponding polyhedron, some additional information related to its height is necessary, such as the altitude of the highest point of the building or, for an approximate value, the number of floors. As this information may be available as attribute information in distinct geographic themes or relational databases, the integration of these distinct data sources becomes a necessity. Additional complexity can be added to this modelling task, as the buildings can have different styles regarding their date of edification, and its architecture can be dependent of the city region where it is located.

The unavailability or inadequacy of base information for a specific modelling task is another kind of problem, that may be overcome with data amplification, and also by introducing randomness in parts of the model where a high level of realism is not relevant. These features are typical of mathematical tools like L-Systems [Prusinkiewicz90], that already are being used in other Computer Graphics works, like the modelling of natural organisms and phenomena [Deussen98] [Lane02] [Prusinkiewicz01], but seldom used in the modelling of urban environments [Parish01].

The main motivation for this work was the development of a modelling system for the expeditious modelling of virtual environments representing real urban areas. This system operates in a semi-automatic mode, based on already available information or additional data to be collected for specific modelling tasks. Due to the large number of elements to be modelled for each urban area, and the variability of the amount and adequacy of available information, the modelling system is also able to deduce additional information, relevant to the modelling process. The shortage of data is no longer an impeding factor for the modelling of virtual environments with high level of detail. High levels of realism can also be easily attained in an iterating workflow, by adding new data at each step or incrementally refining the modelling process.

1.2 The interoperability problem

In the area of Computer Graphics, the interoperability problem has been addressed by the Web3D consortium [Web3D] through the development of the X3D (Extended 3D) [X3D]. Being an evolution of VRML (Virtual Reality Modelling Language) [VRML97], the X3D makes possible the specification of 3D environments and applications in a XML based format. XML is the *de facto* standard for interoperable data representation and publishing, and the whole Internet community is adopting it for the most diverse applications.

The geographic information is probably the type of information with the largest relevance for the urban environments modelling task. This representation, in the form of maps of geometric features and geographic location, are extremely useful to the generation of the three-dimensional models. The OpenGIS consortium [OpenGIS] is an organization devoted to the development of solutions to the interoperability problem in the area of GIS, and has been developing some standards based on XML, like the GML (Geography Markup Language) [GML] or the WFS (Web Feature Service) [WFS] that are rapidly being adopted by the software developers. The integration of this kind of information in the modelling process is of great importance for the development of automatic solutions.

The issues of modularity and reuse of the modelling processes are vital to the speed and efficiency on the generation of an initial solution. Project CONTIGRA [Dachselt02] is clearly an example of how the XML coding of X3D makes possible the integration and reuse of components in the process of three-dimensional applications development.

2. L-SISTEMS

L-systems (acronym for Lyndenmayer systems) are string rewriting techniques developed by Astrid Lyndenmayer in 1968 [Lyndenmayer68] which can be used to model the morphology of a variety of organisms and entities. L-systems are a technology prone to data amplification [Smith84], generating complex structures from small data sets. Another key feature of L-systems is emergence, a process in which a collection of interacting units acquires qualitatively new properties that cannot be reduced to a simple superposition of individual contributions [Taylor92]. These new properties “emerge” from the initial representation.

According to [Prusinkiewicz95], one of the basic concepts in the context of systems L is the module, denoting any discrete constructional unit that is repeated as the system develops. Branches and leaves are an example of this concept, in the modelling of a tree.

The essence of the development at the modular level is captured by a parallel rewriting system that replaces individual ancestor modules by configurations of descendent modules.

All modules belong to a finite alphabet of module types, thus the behaviour of an arbitrarily large configuration of module can be specified using a finite set of rewriting or production rules.

In the simplest case of context-free rewriting, each production replaces a module, called the predecessor, by zero, one or more modules called the successors. In the case of context sensitive L-systems, the applicability of a production does not depend only on the predecessor module, but also on its neighbours.

The productions are applied in parallel, with all modules being rewritten simultaneously at each derivation step. The sequence of strings of modules obtained in consecutive derivation steps, from a predefined initial structure (axiom) is called a developmental sequence.

Parametric L-systems [Prusinkiewicz95] extend the base concept of L-systems, by appending numerical attributes to each symbol. Parametric L-systems operate with parametric words, that is, strings of modules with parameters. In this type of system, the application of a production rule can be dependent of a condition evaluation.

Stochastic L-systems [Prusinkiewicz90] add randomness, through the assignment of a probability value to the production rules.

Our approach to the modelling processes uses parametric and stochastic context sensitive L-systems.

3. THE XL3D MODELLING SYSTEM

To automate the process of creating virtual environments of urban areas we developed a system for three-dimensional modelling, that responds to a modelling specification, by generating a three-dimensional model in X3D format.

The system is based on a procedural approach, where all the modelling processes are specified in a declarative form, with documents based on a XML-Schema [XML-Schema].

Data used for the modelling processes, imported from file directories or databases, should be based in XML formats, and is converted through XSLT [XSLT], into strings of modules. These strings are then concatenated to compose the initial string of data to be used in the modelling processes.

Each modelling process is defined and controlled by a L-system through a set of production rules defined by the user. The initial string of data is then transformed through an iterative process, until a satisfactory solution is obtained.

The final string of data is then interpreted to generate a X3D scenegraph representing the specified three-dimensional model.

3.1 The Modelling Process

Parametric L-systems confer the modelling process a high level of data amplification and emersion, enabling new data inference trough the knowledge that is associated to production rules defined by the user. On the other hand, stochastic L-systems allow the incorporation of randomness in the modelling process, in order to simulate the diversified nature of urban environments.

The first phase in the modelling process consists in the formation of an initial axiom, which is followed by its transformation through the application of production rules and, finally, by the interpretation of the resultant string for generating a scenegraph segment in X3D.

3.1.1 Axiom

The first step in the modelling process is the composition of an initial string of modules, which is called an axiom.

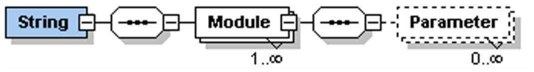

Figure 1 – XML-schema for string of parametric modules

```
<wfs:Edifice fid="Edifice.1">
  <wfs:NFloors>5</wfs:NFloors>
  <wfs:Height>435</wfs:Height>
  <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:Edifice>
```

Figure 2 - Fragment of GML document

```
<XL3D:String>
 <XL3D:Module name=”Origin”>
  <XL3D:parameter>
   524000.08
  </XL3D:parameter>
  <XL3D:parameter>435</XL3D:parameter>
  <XL3D:parameter>
   2116849.16
  </XL3D:parameter>
 </XL3D:Module>
 <XL3D:Module name=”Edifice”>
  <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>
```

Figure 3 - String of parametric modules

The axiom is built from diverse sets of XML documents that are transformed into strings of parametric modules through XSL transformations. The final result of the transformation must follow the XML-schema presented in figure 1.

As an example of the process for creating the axiom, the geographic information representing a building (figure 2) is transformed into a string of parametric modules after the application of a XSLT.

The resulting string (figure 3) is composed by parametric modules containing a name and a sequence of parameters. The axiom can be created from one of these strings or the integration of several others by concatenation or by transformation through XSLT.

3.1.2 Transformation

At this stage, the axiom (figure 4) is processed by a set of production rules that are defined by the user, whose knowledge is used for initial data amplification

```
Origin(524000.084, 435, 211684.916)
Edifice(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”)
```

Figure 4 - Axiom

Each production rule is applied to a specific string of parametric modules, and the notation used in the specification of these production rules is given by:

$$n: \quad L < P > R : cond \xrightarrow{prob} S_1 \ldots S_n \qquad (1)$$

Where ***P*** represents the predecessor module to be identified and replaced by one or more successor modules (***S***); ***L*** represents the left context (modules located in the left side) and ***R*** the right context. The term ***prob*** is the production rule probability, ***cond*** is the associated condition and ***n*** is the number of the production rule.

The production rule applicability is dependent on the existence of a module ***P*** in the current string and on the verification of the associated condition, in the case of a parametric L-system. In the case of context-sensitive L-systems, it is also necessary that modules located on the left and right sides of the predecessor module match the left and right context of the production in order to be applied.

Whenever several production rules can be applied to the same predecessor module, only one is selected randomly based on the associated probability value.

The application of the production rules is done synchronously, after the evaluation and selection process, generating the string of the next derivation step.

This process is repeated successively, until a situation occurs where there is no applicable production, or the maximum number of derivation steps, defined by the user, is reached.

As an example, the axiom presented in figure 4 can be transformed, according to a set of productions, into a string describing the structure of a building. This representation can then be used to drive the generation of the scenegraph of the corresponding three-dimensional model in X3D format.

```
2:Edifice(nfloors,boundary) ─0.25→
  EdificeP1(nfloors, boundary, 1)

3:Edifice(nfloors,boundary) ─0.25→
  EdificeP1(nfloors, boundary, 2)

4:Edifice(nfloors,boundary) ─0.25→
  EdificeP1(nfloors, boundary, 3)

5:Edifice(nfloors,boundary) ─0.25→
  EdificeP1(nfloors, boundary, 4)

6:EdificeP1(nfloors,bound,mat) ─0.5→
  EdificeP2(nfloors,bound,mat,1)

7:EdificioP1(nfloors,bound,mat) ─0.5→
  EdificioP2(nfloors,bound,mat,2)
```

Figure 5 - First production rules

Figure 5 presents a subset of stochastic production rules that permit the random assignment, to the building, of a certain material and type of roof (Figure 6).

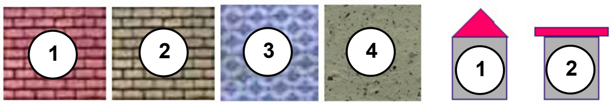

Figure 6 - Facade material and roof type

Another subset of production rules expand the initial two-dimensional contour of the building into a sequence of lines, calculating its perimeter, as well as converting it into a three-dimensional boundary.

```
22:Translation(x,y,z)<EdificeP4
  (nfloors,bound,mat,roof,per,bound3D,
  npts)→
  LOD(0,0,0,250)[ Group
   [Building(mat,bound,nfloors*3,per)]
   [Translation(0,nfloors*3,0)
    [Roof(roof,bound,per)]
    [Gutter(mat,bound3D,per)]]
   [Mopboard(mat,cont3D,per)]
   [ConstFacade(mat,bound3D,npts,
    nfloors*3)]
   [ConstGF(mat,bound3D,per,npts)]
  ]SimpleBuilding(mat,bound,nfloors*3)
  ]
```

Figure 7 - Production rule that structures the building

Figure 7 presents a production rule that creates the base structure of the final scenegraph. The building is transformed into a LOD (Level Of Detail) composed by a detailed model that is visible at distances smaller than 250 meters, and a simpler one, for longer distances.

```
31:
  InstElemFacadeP4(nfloors,mat,x1,y1,

  dx,dy,nbands,band,ang): band>0  ⟶
  InstWindowP1(nfloors,mat,x1+dx
  *(2*band-1)/(2*nbands)),y1+dy*
  (2*band-1)/(2*nbands)),ang)
  InstElemFacadeP4(nfloors,mat,x1,y1,
  dx,dy,nbands,band-1,ang)
```

Figure 8 - Production rule for bands

The detailed model is composed by a building block, a roof with gutter, a ground floor with mopboard and the rest of the facade.

The building boundary is decomposed in facade polygons and, from their width, the production rule 31 (Figure 8) determines the number of bands of windows and balconies.

Figure 9 presents the production rules that instantiate the diverse elements integrating the bands of windows and balconies. The production rules 33 and 34 randomly determines, for each band, if it is composed by windows or by balconies, according to a probability value, that is the double for windows than for balconies. Rule 35 decomposes each band into the set of positions related to the distinct floors of the building where balconies and windows should be positioned. In order to increase the model realism, production rules 36 to 39, make a random selection of the shutter position, open or closed.

The developmental sequence of the L-system terminates after executing all iterations, producing a final string of modules, represented in figure 10. For the sake of legibility, the resulting values where replaced by variables.

33: **InstWindowP1**(nfloors,mat,x,y,ang)
$\xrightarrow{0.67}$
InstWindowP2(nfloors,mat,x,y,ang,0)

34: **InstWindowP1**(nfloors,mat,x,y,ang)
$\xrightarrow{0.33}$
InstWindowP2(nfloors,mat,x,y,ang,1)

35: **InstWindowP2**(nfloors,mat,x,y,a,b)
: nfloors > 1 $\longrightarrow$
[**InstW**(x,3*(nfloors-1)+1.5,y,a,mat,b)]
InstWindowP2(nps-1,mat,x,y,a,b)

36: **InstW**(x,y,z,a,m,b):b=0 $\xrightarrow{0.67}$
Win(x,y,z,a,m)

37: **InstJ**(x,y,z,a,m,b):b=1 $\xrightarrow{0.67}$
Balc(x,y,z,a,m)

38: **InstJ**(x,y,z,a,m,b):b=0 $\xrightarrow{0.33}$
Win2(x,y,z,a,m)

39: **InstJ** (x,y,z,a,m,b):b=1 $\xrightarrow{0.33}$
Balc2(x,y,z,a,m)

Figure 9 - Facade production rules

3.1.3 Interpretation

After producing the final string of modules representing a certain object, it is necessary to interpret it in order to generate the corresponding scenegraph in X3D format.

```
[Translation(x,y,z)
LOD(0,0,0,range)[Group
  [Building(mat,bound,height,perim)]
  [Translation(x,y,z)
   [Roof(type,bound,perim)]
   [Gutter(mat,bound3D,perim)]]
  [Mopboard(mat,bound3D,perim)]
  [Win(x,y,z,a,m)][Win2(x,y,z,a,m)]
  [Win(x,y,z,a,m)][Win(x,y,z,a,m)]
  [Balc(x,y,z,a,m)][Balc(x,y,z,a,m)]
  [Balc(x,y,z,a,m)][Balc2(x,y,z,a,m)]
  [Win2(x,y,z,a,m)][Win2(x,y,z,a,m)]
  [Win2(x,y,z,a,m)][Win(x,y,z,a,m)]
  [Win2(x,y,z,a,m)][Win(x,y,z,a,m)]
  [Win2(x,y,z,a,m)][Win2(x,y,z,a,m)]
  [Balc2(x,y,z,a,m)][Balc(x,y,z,a,m)]
  [Balc(x,y,z,a,m)][Balc(x,y,z,a,m)]
  [Balc2(x,y,z,a,m)][Balc(x,y,z,a,m)]
  [Balc(x,y,z,a,m)][Balc2(x,y,z,a,m)]
  [Win(x,y,z,a,m)][Jan2(x,y,z,a,m)]
  [Win(x,y,z,a,m)][Win2(x,y,z,a,m)]
  [Win(x,y,z,a,m)][Win(x,y,z,a,m)]
  [Win2(x,y,z,a,m)][Win(x,y,z,a,m)]
  [Win(x,y,z,a,m)][Win2(x,y,z,a,m)]
  [Win(x,y,z,a,m)][Win(x,y,z,a,m)]
  [Door(x,y,z,a,m)]]
 SimpleBuilding(mat,bound,height)]
```

Figure 10 - Final string representing a building

Each parametric module is replaced by a segment of a scenegraph, from which a certain number of parameters are instantiated from the arguments values.

The final string of modules is thus interpreted as an hierarchical structure, shaped as a tree, in which the first module (on the left) is the root, and the last is a leaf. In order to allow the creation of a branching structure, an additional set of symbols (brackets - **[** and **]**) delimit each branch.

Prusinkiewicz was a pioneer in the application of L-systems in the area of Computer Graphics. One of his works [Prusinkiewicz86] introduced a new interpretation method, based on the "turtle" metaphor, introduced previously in the LOGO language [Abelson82]. In this approach, a final string is obtained from a L-system and, through a restrict vocabulary of symbols, it is interpreted, adjusting the positioning and orientation of a cursor. This cursor then moves in the three-dimensional space and instantiates 3D primitives along its path.In the current work, a different approach has been used. It is based on the interpretation of a hierarchical structure, shaped as a tree, in order to generate a X3D scenegraph by replacing each parametric module by a prototype. Prototypes are like words from a vocabulary, which the user must create (or import) in order to produce meaningful sentences. The implementation of prototypes expands the X3D PROTO node, allowing the use of expressions for the field definitions.

From the final string representing a building (figure 10), and after interpretation by a set of prototypes, a X3D scenegraph is obtained as a final result. One model is displayed, as an example, in figure 11..

Figure 11 – X3D model of a building

3.1.4 Model Incremental Refinement

The solution obtained from the initial application of a specific L-system does not produce always the best results with a sufficient level of realism, according to the user needs.

In this situation, the user can either incrementally refine the set of production rules, or collect more data to add to the axiom, in order to attain the desired solution.

The system is thus of incremental refinement, enabling the expedite production of an initial model, although with low detail, that can be improved according to the user needs.

3.2 Modelling Specification (XL3D schema)

The whole modelling process is specified in a declarative mode through a XL3D document. A XL3D document is XML based and is created accordingly to a XML-schema specifically developed for this purpose, also named XL3D [Coelho03].

The basic structure of a XL3D document is hierarchical and the root element is the XL3DProject (Figure 12). A XL3D project is composed by a header with authoring information, a set of models comprising the hierarchical structure of entities that recreates the urban environment, modelling procedures that incorporate the relevant knowledge for the modelling process, prototypes that interpret the L-system, and data sources.

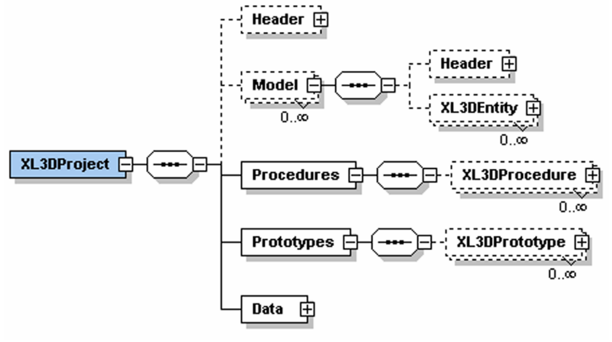

Figure 12 –XL3D Project

The header (HeaderType in figure 13) is intended to provide authoring information to the user, and consists of a title for the element to which is associated, the name and contact information about the authors, version number and links to additional documentation. Due to its structuring character, this element integrates most of the remaining elements of the XL3D schema.

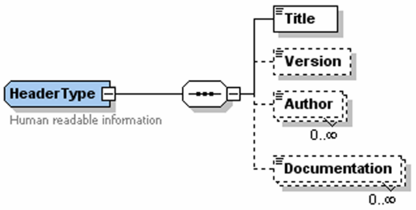

Figure 13 – Header

Each model (Figure 12) is composed by entities, reflecting the complex and structured character of urban environments, according to the subjective perception of the user.

Each entity (XL3DEntityType in figure 14) is a part of the whole model and instantiates modelling procedures to generate 3D models from strings of modules created from diverse data sources. Each entity can be composed by other entities whose model can be placed in any specific position of the entity's scenegraph and can be submitted to a coordinate transformation.

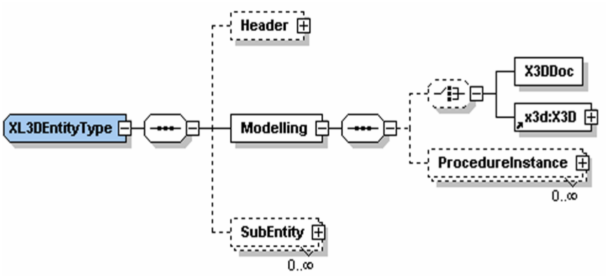

Figure 14 – Entity

As an example, a city model can be composed by some entities like the road network, the vegetation and the buildings. In turn, some entities like the pavement, the sidewalks and signposts, could also compose the road network. Each entity instantiates a set of modelling procedures that generate three-dimensional models from strings of modules obtained from diverse data sources.

The axiom is composed from one or more data sources. These data sources are combined by concatenation or the application of a XSLT.

The modelling procedures (XL3DProcedureType in Figure 15) stand for the nuclear element of the modelling process, and are based on production rules that control the L-system that models specific parts of each entity.

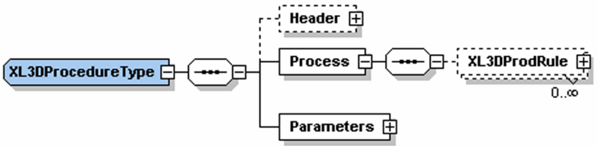

Figure 15 – Modelling procedure

The modelling procedures are defined as reusable components that may be instantiated for the modelling of distinct entities, or extended to build new modelling procedures.

Each modelling procedure contains a set of production rules that transform an initial string of modules, called the axiom. The production rules (XL3DproductionType in figure 16) are specified according to equation (1) regarding context-sensitive L-systems, with parametric and stochastic extension.

The resultant string of modules is interpreted in order to create a scenegraph, through the instantiation of a set of prototypes.

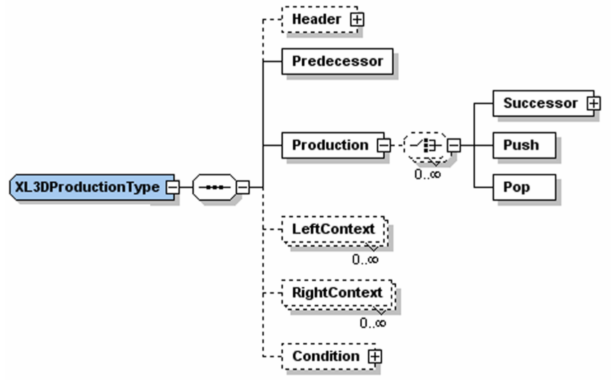

Figure 16 – Production rule

Prototypes (XL3DPrototypeType in figure 17) are segments of a X3D scenegraph that can be instantiated through the interpretation of the final string of modules obtained from the application of modelling procedures.

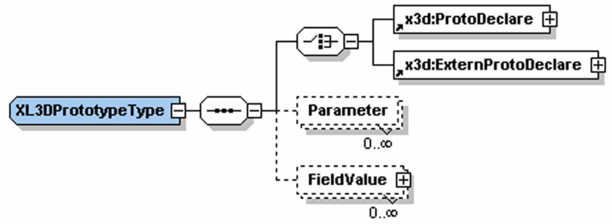

Figure 17 – Prototype

Data sources (DataSourceType in figure 18) contain references to data stored in XML documents, either as files in directories or as response to the invocation of a Web service. Data are converted from its original format to a string of modules, through XSLT.

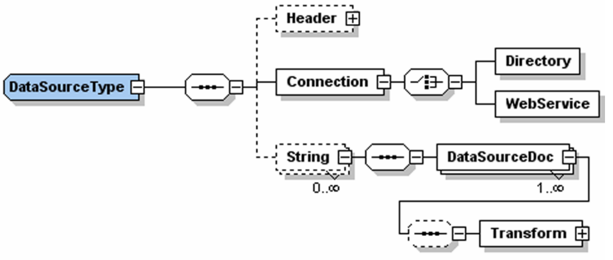

Figure 18 – Data source

4. RESULTS

4.1 The Modelling Process

The process of generating the model of a building model from an initial axiom composed by two modules was exemplified in section 3.1. After data amplification, forty six modules composed the final string, obtained by the modelling process, as a representation of the building. As result, a X3D document containing 277 modes was generated after interpretation. These results clearly demonstrate the potential of data amplification for the generation of complex models from small data sets.

Good results are also being obtained from the emergence concept, as it can be seen in the X3D scene in figure 11: several details like balconies and windows geometry, the roof shape and the building colour and textures are being used but were not obtainable only from the horizontal contour contained in the original axiom.

The effect of randomness is also visible in figure 11, by the different positions of the shutters in balconies and windows, useful to reflect the typical nature of an inhabited building.

4.2 The XL3D Modelling System

The process that was applied to just one element in the above example, can also be applied to axioms containing a large number of elements, enabling the generation of extensive urban environments without any additional user effort.

Based on this modelling process, we developed a system, called XL3D that automatically generates three-dimensional models from a modelling specification, declared in a XL3D document.

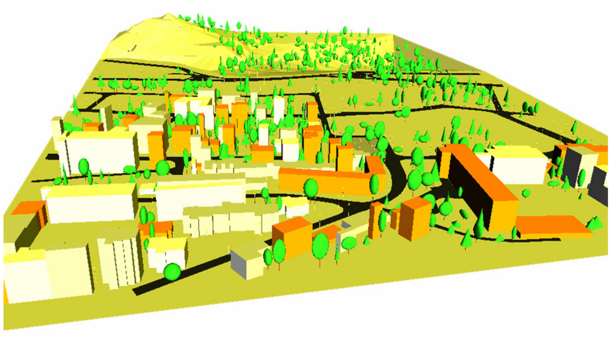

Figure 19 – Simple model of urban environment

Extensive virtual environments corresponding to urban areas (figure 19) can be modelled by this system, starting from small data sets (figure 20) and using a simple modelling specification.

Figure 20 - Available geographic information

The initial solution can then be incrementally refined until the desired level of detail and realism is obtained. Figure 21 shows an initial model, characterized by simplified models in buildings and trees. Nevertheless, the geometry generated permits to recognize the main objects, namely lights in the streets. Figure 22 presents a refined scene, obtained by a better set of production rules, namely in the generation of buildings and trees. It can be seen that now, buildings present textures and new 3D issues like balconies and windows, as well as different roofs.

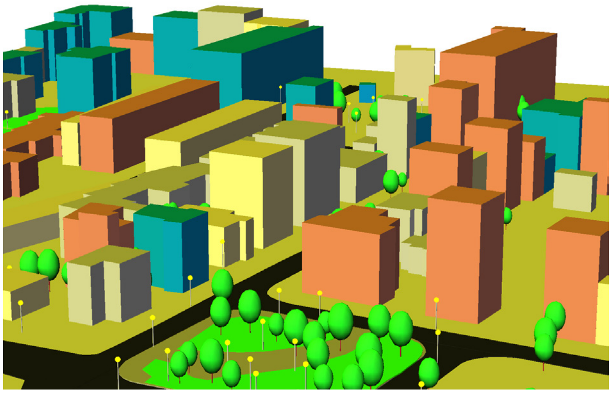

Figure 21 – Initial model of an urban environment

Also the trees are now represented by appropriate billboards (that could even be distinct, accordingly to different specimens) and the terrain and pavements are covered with textures, adequate to their nature.

The randomness inherent to the modelling process, altogether with a small initial dataset can produce different results, which reinforces the importance of the amount and quality of the initial data. Figures 22 and 23 present a similar level of detail, but Figure 23 is more realist because buildings heights were added to the initial data. The scene could be even ameliorated with the inclusion, in the initial data, of generic information about facades.

Figure 22 – Refined model of an urban environment

Due to the data amplification and emergence features that characterize this system, distinct models can be obtained from the same initial data and modelling specification (figure 23).

Using sets of modelling procedures and associating them to some axioms representing diversified information enables the user to model the distinct elements that compose any urban environment.

The modelling processes specified for a certain case can easily be reused and applied to other problems with similar characteristics, extending the set of production rules.

Figure 23 – Alternate model of an urban environment

5. CONCLUSIONS

The modelling system developed can be used with advantages in the generation of three-dimensional models of urban environments, reducing both the costs and the duration of the process. The obtained tree-dimensional model can incrementally be improved, through the refinement of the production rules that control the modelling process, as well as through the addition of more data.

The system can produce three-dimensional models with high level of detail and realism, as long the available information is vast and has quality. However, even in the cases where such information is scarce, the system is capable of generating an initial solution that, although with a lesser level of realism, maintains an acceptable level of detail via the incorporation of some randomness.

In the specific case of some complex elements of the model, like monuments and emblematic buildings, the XL3D system cannot attain the adequate level of realism. These elements must therefore be carefully modelled with conventional modellers and later integrated in the modelling process.

The definition of the set of production rules to the modelling of a certain entity is not trivial, but a rigorous and iterative process. However, the use of parametric modelling procedures, as well as the easiness to extend a specific process to model a distinct entity prevents that difficulty and enable the reuse of many of the modelling processes.

The XL3D modelling system is now being converted to a Web Service as a modeller of urban environments, available to public. The main advantage will be to permit the invocation by other applications, in a distributed architecture and in an automatic fashion, to generate 3D models of urban scenes or parts of scenes. Furthermore, this architecture will permit, from the system invocation to other Web Services, the acquisition of other and diverse initial information, properly updated, with benefits to the models realism, as well as to the frequency of the scene update maintenance (in the limit, a permanent synchronisation between the virtual and the real models). Given the advances in the specification of Web services for the publishing of geographic information, like the WFS [WFS], the access to the result of advanced queries to the database and spatial operations will be simplified.

An example of this architecture relevance, is a mobile real time navigation system [Pinto03] that is being developed and for which it is intended to integrate three-dimensional models of the surrounding environment. Since the trajectories are determined in real time, as a reply to a user request, these models should be generated dynamically from a web service.

A weakness of the presented system is the way the XL3D projects are specified, since this is done by means of generic tools to edit XML. A new work, already started, is developing an adequate edition tool, intuitive and easy to use, that puts a special emphasis on the creation of the production rules.

6. ACKNOWLEDGEMENTS

This work is partially supported by POSI, the European Union and FEDER through the project POSI/CHS/48220/2002 entitled "3D4LBMS - Three-dimensional Urban Virtual Environment Modelling for Location Based Mobile Services".

7. REFERENCES

[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 ICCC/IFIP International Conference on Electronic Publishing*, pp. 278-287, 2003.

[Dachselt02] Dachselt R., Hinz M., Meiβner 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.: Creation and management of urban 3D scene complexity in immersive virtual environments. *Proceedings of the 10th Portuguese Meeting on Computer Graphics*, pp. 165-174, 2001 (in Portuguese).

[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 Artifitial 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 Tranformations (XSLT). `<http://www.w3.org/TR/xslt11/>`