3D Modelling of Large Urban Scenes from Diverse Sources of Information

ANTÓNIO FERNANDO COELHO[1]; ANTÓNIO AUGUSTO SOUSA[2,3]; FERNANDO NUNES FERREIRA[2]

[1] Universidade de Trás-os-Montes e Alto Douro
Quinta de Prados, 5000-911 Vila Real, Portugal
e-mail: acoelho@utad.pt;
[2] Faculdade de Engenharia da Universidade do Porto
Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
e-mail: aas@fe.up.pt; fnf@fe.up.pt
[3] INESC Porto
Campus da FEUP
Rua Dr. Roberto Frias, 4200-465 Porto, Portugal

The complex and extensive nature of urban environments creates difficulties to the task of generating virtual models. Thus a great effort in terms of human resources, time and money is needed. Nevertheless a large number of professionals and institutions devout efforts to gather and analyse data from these urban environments. As data is usually stored in a digital format, it becomes a valuable asset to incorporate it in the modelling process of virtual environments. This paper presents a three-dimensional modelling system with interoperable access to data in diverse formats and digital support, drived by an L-system based modelling process that automatically generates initial solutions for virtual environments, which can be incrementally improved.

Keywords: 3D Modelling; L-systems; Virtual Reality; XML; X3D; XSLT; Computer Graphics; Geographic Information Systems

INTRODUCTION

Virtual environments from urban areas are an important tool to a diverse set of applications, like virtual tourism, publicity, education and cultural heritage, among others. The virtual city of Lisbon[1] is one example of the use of virtual environments and virtual reality in urban planning. The virtual reconstruction of ancient city of Sagalassos[2] is another example in the area of cultural heritage.

However, generating 3D Models of large urban environments poses a large number of computer graphics problems, as the amount of information needed to create realistic models increases with size and complexity. Although some models have been created by individual modelling of each element in the urban environment, semi-automatic methods have proven to be much more effective by reducing time, costs and human effort[1,3]. There are commercial applications that provide some automatic procedures for modelling, from geographic information systems[4]. But these functionalities are usually not flexible enough to provide a realistic solution to the modelling of urban environments and there isn't also the possibility to extrapolate new attributes from the existing data.

The complex nature of urban environments is prone to be studied by a diverse set of professionals from geographers to economists, architects to sociologists, historians to archaeologists, etc. The information collected and gathered in the form of documents or databases can be used to drive the modelling process. Usually this information is not enough to, by itself, generate the model, and must be submitted to a process of data amplification, either by providing knowledge about the environment, or including some randomness in parts of the model where information is coarse and a great level of realism is not essential. Realism can be attained in a later phase, by adding more data or refining knowledge, and in some important elements of the model like cultural heritage and relevant infrastructures, adding detail by manual means.

L-systems (acronym for Lyndenmayer systems) are string rewriting techniques developed by Astrid Lyndenmayer in 1968 which can be used to model morphology of a variety of organisms and entities[5]. A lot of work has been done since then using L-systems to model natural organisms and phenomena[6,7], but only a few to model man-made infrastructures[8].

L-systems are a technology prone to data amplification, generating complex structures from small data sets[9]. 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[10]. Randomness can also be attained by using stochastic L-systems[11].

We present a system for the modelling of large urban environments, based on information stored in diverse formats and digital support. Integration of data is done by using Extensible Markup Language (XML)[12], a standard for interoperability. The modelling process is implemented by a procedural approach based on L-Systems, and its specification is based on declarative XML documents, for which a XML-schema was developed. Among others, several advantages in using XML are platform independence, standardization, interoperability with other Internet standards, availability of tools and document processing using the Document Object Model (DOM)[13] or Extended Stylesheet Language Transformations (XSLT)[14].

THE 3D MODELLING SYSTEM

To automate the modelling process to generate virtual environments of urban areas, we have developed a system based on a client-server architecture, that responds to a modelling specification by generating a 3D model in the Extensible 3D format (X3D)[15]. The specification of the modelling process is based on a declarative document approach for which XL3D, a XML based format, was developed.

To drive the modelling process, data from diverse sources and digital support can be used, as long as they are in a XML or text based format. The data sources can be related to geographic information systems, to municipality alphanumeric databases, to historic archives or to directories of documents, among others. This restriction does not limit however de applicability to the majority of data sources, as XML is being adopted by most software developers as an interoperability standard for data exchange and publishing, and text is also a common export format. These data sources are associated with XSL transformations that define how to structure the relevant data into a string of modules, suitable for the modelling process.

The modelling process is based on L-systems, and data is transformed from an initial string of modules, by a set of production rules into a final string describing the desired entity. This final string is interpreted using prototypes to generate the three-dimensional model in X3D. Besides the obvious advantages of X3D being a XML-based standard, and although it has not yet been adopted as an international standard, recent developments by the Web3D consortium[16] give us the idea that it is the probable successor of the Virtual Reality Modelling Language (VRML)[17]. Also the easy conversion between X3D and VRML by using XSL transformations will smooth the transition process, providing mature VRML authoring tools to the new standard.

The combination of interoperable access to diverse data sources and the data amplification and emergence features of L-systems constitute the enabling conditions to the automation of the modelling processes.

INTEROPERABLE ACCESS TO DIVERSE SOURCES OF INFORMATION

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. Either as a file format or as information exchange with a Web service, large amounts of data are being represented by XML based formats and a great number of standards are being regulated by the World Wide Web Consortium[18].

All information related to urban environments, or to the elements that compose them, are a valuable asset to the modelling process. Surely most of this data are based on files, databases or web services that either are natively XML, text based documents, or it is possible to export them to these formats.

From these data sources, geographic information is clearly the most valuable data for the modelling process of urban environments, since it provides data about features in the environment from a 2D representation: a map. Data such as geometric attribute information and geographic location are extremely useful for shaping the components of the virtual environment. As geographic information is gathered by distinct institutions, which normally use different geographic information systems (GIS), difficulties arise to integrate these data. The OpenGIS consortium[19] has been developing some standards based on XML, like the GML (Geography Markup Language)[20] or the WFS (Web Feature Service)[21] that are rapidly being adopted by the software developers.

Interoperable Access to data is provided by a XML-Schema, developed in order to represent almost any type of information as well as being suitable for the modelling process. This XML-Schema, called XL3DString (figure 1), represents a string of parametric modules, based on the L-system concept of modular development.

In order to be used as a data source for the modelling process, data must be transformed to this XML-Schema from their native XML or text based format by using a XSL transformation.

Each data source has a group of links to documents containing the original data, and also associated XLS transformations, that serialize the information that is pertinent to the modelling procedure, to be formatted in the XML-Schema developed.

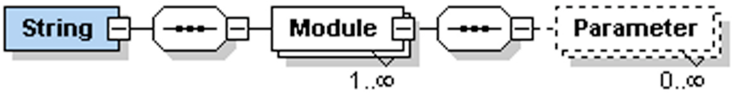

FIGURE 1 – XL3DSTRING SCHEMA DEFINITION

As an example, table 1.a shows a segment of a GML document describing an element of an edifice theme, containing the horizontal contour of the buildings, and some attribute information, like the number of floors or the building height.

By applying a XSL transformation, this segment is transformed in the document of table 1.b according the XML-Schema developed.

THE MODELLING PROCESS

The modelling process is based on context-sensitive L-systems with parametric and stochastic extensions.

One of the fundamental concepts of L-systems is the module, denoting a discrete constructional unit that is repeated as the system develops. Parametric L-systems extend this concept by assigning numerical attributes to each module.

TABLE 1 - EXAMPLE OF INTEROPERABLE ACCESS

A) GML

```
<wfs:Edifice fid=”Edifice.1”>
  <wfs:Floors>5</wfs:Floors>
  <wfs:Height>435</wfs:Height>
  <wfs:Geometry2>
   <gml:Polygon srsName=”Port:ugal”>
    <gml:outerBoundaryIs>
     <gml:LinearRing>
      <gml:coordinates>
       524000.08, 211684.92
       524004.66, 211684.52
       524005.73, 211695.96
       524007.28, 211696.03
       524007.61, 211703.81
       524007.61, 211703.81
       524000.08, 211704.47
      </gml:coordinates>
     </gml:LinearRing>
    </gml:outerBoundaryIs>
   </gml:Polygon>
  </wfs:Geometry2>
</wfs:Edifice>
```

B) STRING OF PARAMETRIC MODULES

```
<XL3D:String>
 <XL3D:Module name=”Origin”>
  <XL3D:parameter>
   524000.08
  </XL3D:parameter>
  <XL3D:parameter>
   435
  </XL3D:parameter>
  <XL3D:parameter>
   211684.92
  </XL3D:parameter>
 </XL3D:Module>
 <XL3D:Module name=”Edifice”>
  <XL3D:parameter>
   5
  </XL3D:parameter>
  <XL3D:parameter>
   0.00 0.00,
   45.72 -3.97,
   56.43 110.42,
   71.96 111.10,
   75.24 188.97,
   75.21 188.97,
   0.00 195.51,
   0.00 0.00
  </ XL3D:parameter>
 </XL3D:Module>
</XL3D:String>
```

A parallel rewriting mechanism, replaces individual predecessor modules by configurations of successor modules. This behaviour is specified using a finite set of rewriting rules or productions.

In context-sensitive L-Systems, productions are dependent not only on the predecessor module, but also on it’s left and right neighbour modules. In parametric L-systems productions can be also dependent on conditions evaluated from parameters values of the modules.

Stochastic L-systems extend the basic concept by introducing randomness, allowing more than one production to be applicable at the same time, and optionally assigning a probability value to each production.

The L-system’s development starts with an initial string of modules, called an axiom that is constructed as a concatenation of modules strings defined from the data sources described in the previous section.

A set of productions defined by the user and representing his knowledge of the problem, transforms the axiom, over a sequence of steps, into a final string representing the structure to be interpreted as a model.

As an example and based on the data source represented in table 1 we obtain the axiom represented in table 2.

The axiom is transformed by a set of 37 productions in the final string represented in table 3. By the sake of simplicity and readability, the parameters values of the modules were not represented, replacing them by their names. The symbols [and] represent parallel branches of a tree-like hierarchical structure.

TABLE 2 - EXAMPLE OF AN AXIOM

```
Origin(524000.08, 435, 211684.92)
Edifice(5,"0.00 0.00, 45.72 -3.97, 56.43 110.42, 71.96 111.10, 75.24 188.97, 75.21
188.97, 0.00 195.51, 0.00 0.00")
```

The comparison between the two strings of parametric modules, demonstrates the L-systems features of data amplification and emergence. The simple horizontal contour of a building is automatically transformed into a description of a building with roof, façade composed with windows, balconies and a door, a mopboard and a gutter. From an initial string of 2 modules, a coherent final string composed by 44 modules was obtained.

TABLE 3 - FINAL STRING REPRESENTING A BUILDING

```
Translation(x,y,z)
 [Building(material,boundary,height)]
 [Façade
  [Window(x,y,z,angle,material)][Window2(x,y,z,angle,material)]
  [Window(x,y,z,angle,material)][Window(x,y,z,angle,material)]
  [Balcony(x,y,z,angle,material)][Balcony(x,y,z,angle,material)]
  [Balcony(x,y,z,angle,material)][Balcony2(x,y,z,angle,material)]
  [Window2(x,y,z,angle,material)][Window2(x,y,z,angle,material)]
  [Window2(x,y,z,angle,material)][Window(x,y,z,angle,material)]
  [Window2(x,y,z,angle,material)][Window(x,y,z,angle,material)]
  [Window2(x,y,z,angle,material)][Window2(x,y,z,angle,material)]
  [Balcony2(x,y,z,angle,material)][Balcony(x,y,z,angle,material)]
  [Balcony(x,y,z,angle,material)][Balcony(x,y,z,angle,material)]
  [Balcony2(x,y,z,angle,material)][Balcony(x,y,z,angle,material)]
  [Balcony(x,y,z,angle,material)][Balcony2(x,y,z,angle,material)]
  [Window(x,y,z,angle,material)][Window2(x,y,z,angle,material)]
  [Window(x,y,z,angle,material)][Window2(x,y,z,angle,material)]
  [Window(x,y,z,angle,material)][Window(x,y,z,angle,material)]
  [Window2(x,y,z,angle,material)][Window(x,y,z,angle,material)]
  [Window(x,y,z,angle,material)][Window2(x,y,z,angle,material)]
  [Window(x,y,z,angle,material)][Window(x,y,z,angle,material)]
  [Door(x,y,z,angle,material)]]
 [Mopboard(material,boundary,perimeter)]
 Translation(0,height,0)
 [Roof(material,points,perimeter)]
 [Gutter(material,boundary,perimeter)]
```

INTERPRETATION AND GENERATION OF THE X3D MODEL

The final string of parametric modules (like the one in table 3) cannot, by itself, generate the final X3D model. It must be interpreted to generate a valid X3D document.

In X3D, a three-dimensional model is called a scene and its definition is based on a tree-like hierarchical structure called a scenegraph. In our modelling process, the last phase consists on replacing each module by a segment of a X3D scenegraph, called a prototype. The modules' parameters are instantiated as arguments of the prototype. The prototypes are user defined and its specification should be made prior to the definition of the L-system productions, in order to correctly interpret the final string of modules.

Prototypes are like words from a vocabulary, which the user must create (or import) and be conscious, to be able to produce meaningful sentences.

From the final string of modules in table 3 and after interpretation we obtain the X3D scenegraph in figure 2.

FIGURE 2 - X3D MODEL OF A BUILDING

XL3D SCHEMA DEFINITION

An XML-schema, named XL3D, was developed to support the modelling process specification.

XL3DPROJECT

The root element of any XL3D document is the XL3Dproject (figure 3). A XL3D project is composed by a header, models, procedures, prototypes and data sources.

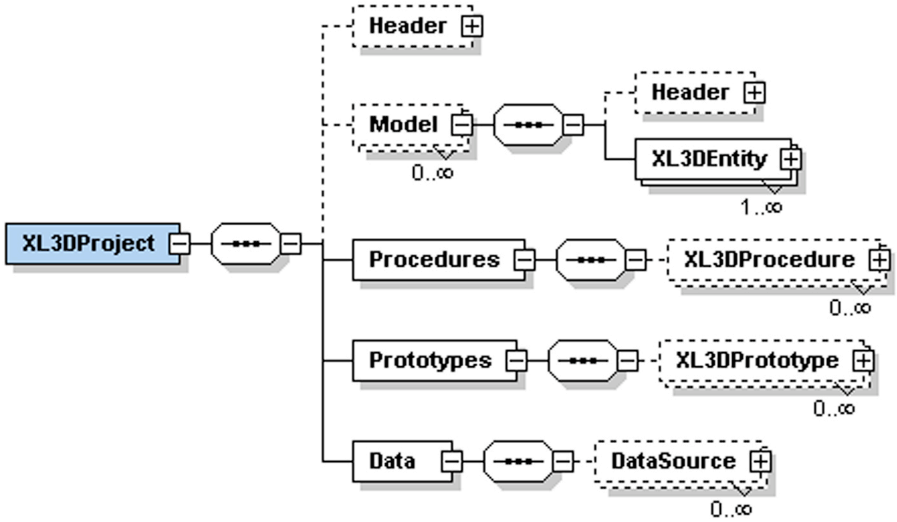

FIGURE 3 - XL3DPROJECT

The header includes information to the user about the project, and the models comprise the hierarchical structure of entities that recreate the urban environment from the user conceptual description.

Each project may contain a set of models, each one composed of Entities that reflect the complexity of the urban environment as viewed by the user and outputting the result from the modelling process to a X3D document.

HEADER

The header (figure 4) is intended to provide authoring information to the user, and consists of a title naming the element to which is associated, the name and contact information about the authors, version number and also some links to additional documentation.

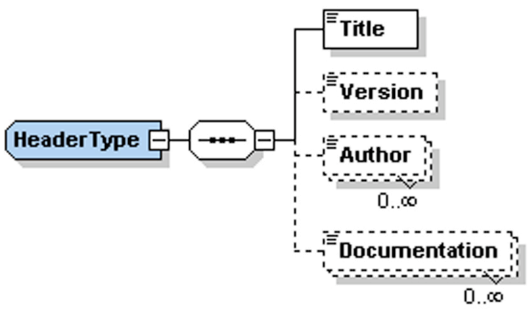

FIGURE 4 - HEADERTYPE

XL3DENTITY

Each entity 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 scene graph and can be submitted to a coordinate transformation.

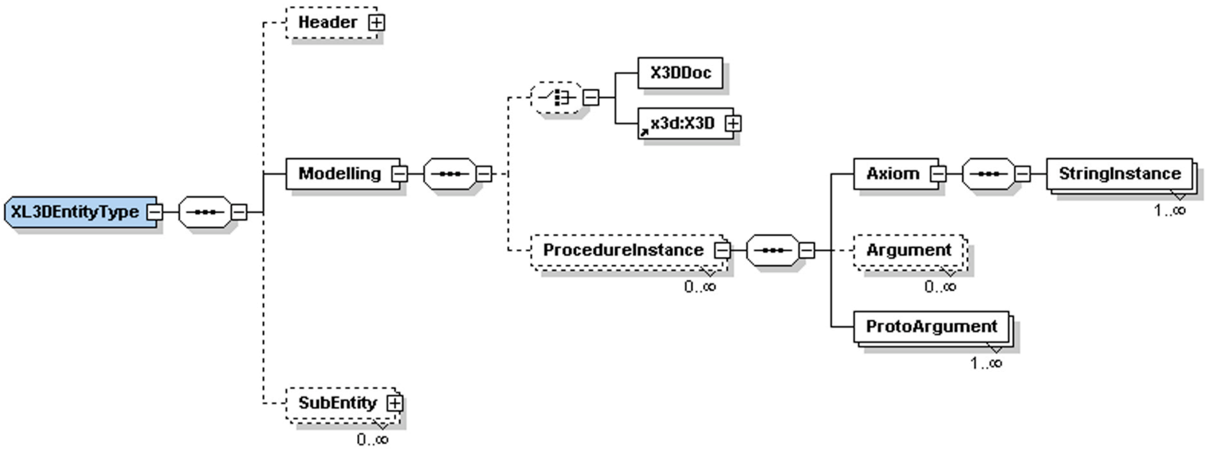

FIGURE 5 - XL3DENTITYTYPE

An element of type XL3DEntityType (figure 5) is composed by a header, a modelling element and a set of sub-entities.

The modelling element is composed by a X3D segment of a scenegraph, either incorporated in the document, or in another one, and a sequence of ProcedureInstance elements.

A ProcedureInstance element is composed by an axiom and a sequence of Arguments and ProtoArguments, that instantiate respectively, values of parameters and the prototype definitions. The Axiom element concatenates a sequence of elements of type StringInstance that reference string definitions in data sources. Elements of type Argument define values to be passed to parameters of the modelling procedures and the elements ProtoArgument reference prototypes to interpret a specific module of the final string.

XL3DPROCEDURE

Modelling procedures are the core of the modelling process by specifying the L-system that models specific parts of each entity. A modelling procedure (figure 6) is composed by a header, a modelling process description, a set of parameters that can be instantiated as arguments of the parametric modules, and prototype parameters that specify which modules will be interpreted by prototypes defined afterwards in a ProcedureInstance element. The modelling procedures are defined as reusable components that can be instantiated to model distinct entities.

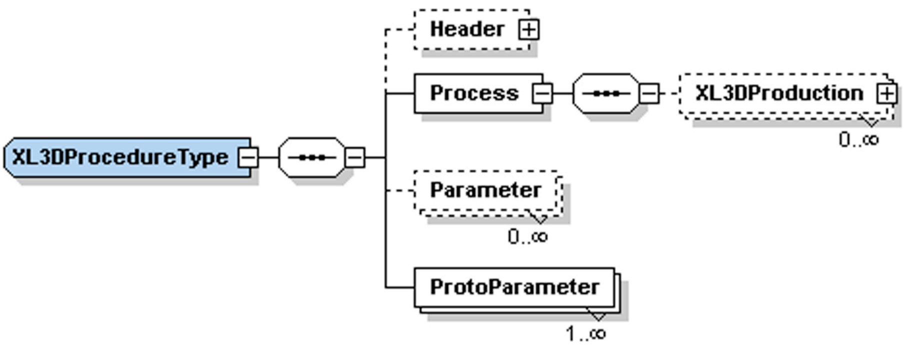

FIGURE 6 - XL3DPROCEDURETYPE

XL3DPRODUCTION

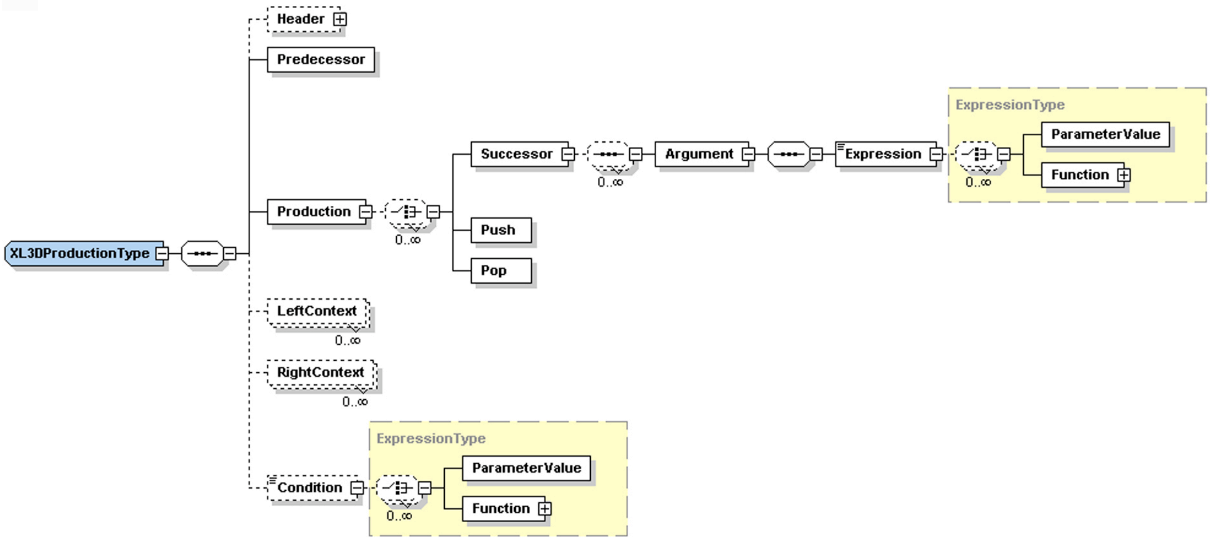

FIGURE 7 - XL3DPRODUCTION

The modelling process is specified by a sequence of productions of type XL3Dproduction.

The XL3DProductionType (figure 7) defines a production of a L-system. It's composed by a predecessor module to be replaced, a sequence of successor modules, in which Push and Pop represent the branch operators symbolized by the brackets, two sequences of left and right context modules and a condition. A probability value is also defined as a type attribute.

Both the arguments of the successors and the condition are expressions. These expressions may contain constants and operators, and also references to parameters or functions.

XL3DPROTOTYPE

Prototypes are described by the XL3Dprototype (figure 8) type definition and are parametric segments of X3D scenegraphs that can be passed as arguments to the modelling procedures to be instantiated. Thus, a prototype encapsulates the complexity of X3D by providing the user with a simpler interface, based on a single parametric module that can be easily instantiated with arguments related to the nature of the element. Although X3D has some prototype nodes, its parameters are too closely related to the nature of the X3D model.

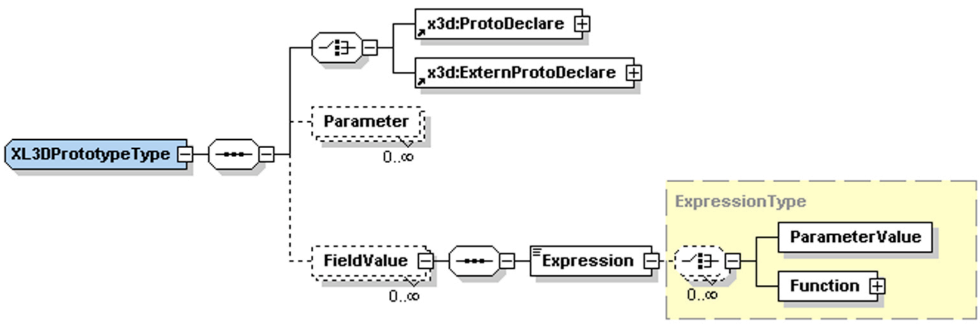

FIGURE 8 - XL3DPROTOTYPETYPE

XL3D prototypes are composed of a an X3D prototype declaration, a sequence of parameters and a sequence of field values. X3D prototypes can be locally defined by using x3d:ProtoDeclare, or externally defined in another document with x3d:ExternProtoDeclare. When instantiating a prototype, some values must be attributed to the prototype fields. These values are obtained by calculating expressions using the XL3D parameters defined in the module. These expressions translate the conceptual description of a certain element from an urban environment to a three-dimensional model description.

DATASOURCE

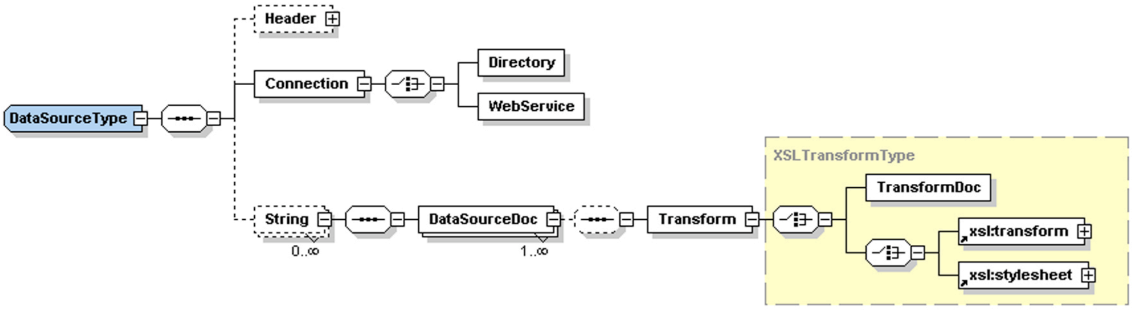

FIGURE 9 - DATASOURCETYPE

Data sources contain references to information of a single type, either from a group of files in a directory or a group of documents returned by a web service. The data is converted from XML to a string of parametric modules.

Thus, data for modelling must be in XML format or should be converted to XML by the use of the exporting capabilities of the application supporting the data, or by converting to a text-based format and then, by using XSL transformations, converting again to XML.

DataSourceType elements (figure 9) are composed by a header, a connection element and a sequence of strings. A string element is composed of a sequence of documents optionally associated to a XSLT transform, either embedded or in another document.

RESULTS

A comparison between table 2 and table 3 demonstrates the effect of data amplification, by transforming an axiom of two modules to a final string of 44 modules, featuring a ratio of 22. Also clear is the effect of emergence. When analysing the X3D scene in figure 2, we can observe some elements that could not be derived from just the horizontal contour contained in the original axiom.

The final string constitutes a coherent representation of the user conceptual description to obtain a 3D model of a building.

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

XML is the enabling technology to the integration of data in different formats. Interoperability is achieved by the use of XSLT technology to transform the different formats into a simple common format, defined by the XL3DString schema. This format represents information as a string of parametric modules, ideally suited to drive the L-system based modelling process.

The system has clear advantages for the modelling of large urban areas like the one seen in figure 10.b that was automatically generated from the geographic information represented in figure 10.a, using very simple modelling processes. The user can, after this, ameliorate the model, specifying more refined modelling processes, or even including manually modelled objects.

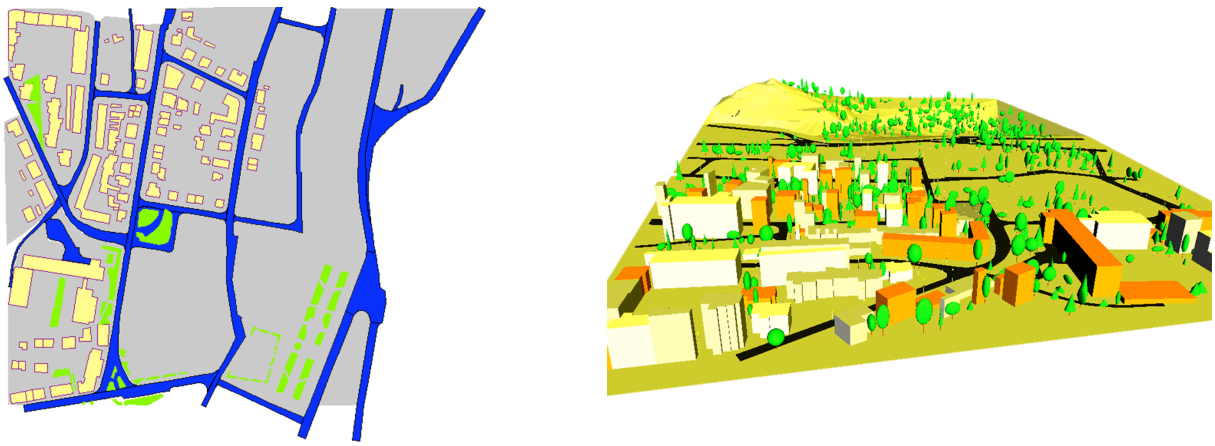

FIGURE 10 – GENERATING A VIRTUAL ENVIRONMENT FROM GEOGRAPHIC INFORMATION
A) GEOGRAPHIC INFORMATION **B) VIRTUAL ENVIRONMENT**

CONCLUSIONS AND FUTURE WORK

The modelling system developed can be used with clear advantages to model large urban environments, by reducing the time to generate an initial solution. In the case the initial solution might not be completely satisfying, it can be incrementally improved, either by collecting and gathering new data, or by refining the modelling process.

The system can be applied with high level of detail and realism in environments where a large amount of information (namely geographic) is available, but can also be used to model environments where the information is sparse. Higher level of detail can also be achieved in this last case, due to the data amplification feature of L-systems, but without the same level of realism, as more randomness is introduced.

This system can be very useful for modelling existing urban environments, because usually, there exist a large amount of digital information. Applications are diverse, from virtual tourism to urban planning, among others.

In spite of its interest for the cultural heritage, the system presents some limitations, namely in some special buildings, or where manual intervention is needed to achieve a highly detailed and realistic model. Nevertheless, these detailed models can be instantiated afterwards in the modelling process.

Reconstruction of such environments is generally an iterative process where more data is collected and analysed to incrementally improve knowledge, at each step. The system developed could be integrated in this cycle as a visual testbed for studying these cultural heritage sites.

The system architecture is being developed to become a Web service providing automatic 3D modelling to client applications. This can be very useful, for instance in location based mobile systems (LBMS), where segments of 3D models must be dynamically generated and transmitted to the mobile platform based on location.

Future developments of this system will address the aspect of authoring tools to create XL3D schema based documents in an intuitive graphical representation. Also some client applications are being developed to create information useful to the modelling process from real imagery of the urban environments to be modelled.

NOTES AND REFERENCES

1 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, Lisbon, 2001, pp. 165-174.

2 ______. *Virtual Salagassos*. [online] [cited 2003-05-15] Available from: <http://www.esat.kuleuven.ac.be/sagalassos/>.

3 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.

4 ESRI. *ArcGIS 3D Analyst*. [online] [cited 2003-05-15] Available from: <http://www.esri.com/software/arcgis/arcgisxtensions/3danalyst/index.html>.

5 LINDENMAYER A. *Mathematical models for cellular interaction in development.* Parts I and II. Journal of Theoretical Biology, 18:280–315, 1968.

6 PRUSINKIEWICZ P.; MUENDERMANN L.; KARWOWSKI R.; LANE B. *The use of positional information in the modeling of plants*. Proceedings of SIGGRAPH 2001, Los Angeles, 2001, pp. 289-300.

7 LANE B.; PRUSINKIEWICZ P. *Generating spatial distributions for multilevel models of plant communities*. Proceedings of Graphics Interface 2002, Calgary, 2002, pp. 69-80.

8 PARISH Y.; MULLER P. *Procedural Modeling of Cities*. Proceedings of SIGGRAPH 2001 Los Angeles, 2001, pp. 301-308.

9 SMITH A. *Plants, fractals, and formal languages*. Proceedings of SIGGRAPH '84, Mineapolis, 1984, in ComputerGraphics, 18, 3 (July 1984), New York: ACM SIGGRAPH, 1984, pp. 1-10.

10 TAYLOR C. *"Fleshing out" Artificial Life II*. In Langton C.; TAYLOR C., FARMER J.; RASMUNSSEN S., ed. Artificial Life II, Redwood City: Addison –Wesley, 1992, pp. 25-38.

11- PRUSINKIEWICZ P.; LINDENMAYER A. *The Algorithmic Beauty of Plants*. Springer, 1990.

12 W3 Consortium. *Extensible markup Language (XML)*. [online] [cited 2003-05-15] Available from: <http://www.w3.org/XML/>.

13 W3 Consortium. *Document Object Model (DOM)*. [online] [cited 2003-05-15] Available from: <http://www.w3.org/DOM/>

14 W3 Consortium. *XSL Transformations (XSLT)*. [online] [cited 2003-05-15] Available from: <http://www.w3.org/TR/xslt11/>

15 WEB3D Consortium. *X3D: The Virtual Reality Modelling Language – International Standard ISO/IEC 14772:200x*. [online] [cited 2003-05-15] Available from: <http://www.web3d.org/TaskGroups/x3d/specification/>

16 WEB3D Consortium. *Web3D Consortium*. [online] [cited 2003-05-15] Available from: <http://www.web3d.org/>
17 WEB3D Consortium. *The Virtual Reality Modelling Language – International Standard ISO/IEC 14772-1:1997*. [online] [cited 2003-05-15] Available from: < http://www.web3d.org/technicalinfo/specifications/ISO_IEC_14772-All/index.html/>
18 W3 Consortium. *W3C - World Wide Web Consortium*. [online] [cited 2003-05-15] Available from: <http://www.w3c.org/>
19 OPENGIS Consortium. *The OpenGIS Consortium, Inc.*. [online] [cited 2003-05-15] Available from: <http://www.opengis.org/>
20 OPENGIS Consortium. *Geography Markup Language (GML)*. [online] [cited 2003-05-15] Available from: <http://www.opengis.net/gml/>
21 OPENGIS Consortium. *Web Feature Service*. [online] [cited 2003-05-15] Available from: <http://www.opengis.org/techno/RFC13.pdf>