Exploring CityEngine as a Visualisation Tool for 3D Cadastre

Alexandra RIBEIRO, José-Paulo DUARTE de ALMEIDA, Portugal and Claire ELLUL, UK

Key words: 3D Cadastre, 3D Visualisation, ESRI CityEngine, Procedural Modelling, CGA Shape Grammar

SUMMARY

3D visualisation is a graphical way to identify and spatially communicate the complexity of a large number of real life situations of overlapping and encroachments in 2D or 3D land and property interests (e.g., buildings with complex architecture, infrastructures above or below Earth surface, natural resources and corresponding rights). Currently, several 3D visualisation applications and cadastral prototypes have been developed around the world. However, they still require maturation and validation by the users before being able to be used in real life situations. Thus, research on 3D cadastral visualisation needs more investigation.

The 3D modelling of urban environments utilizes 3D visualisation systems. If these systems can somehow be reutilized in the 3D cadastre context, associate costs might be lower by building a system out of scratch. One of those systems is the ESRI CityEngine. This work proposes the evaluation of CityEngine's suitability as a 3D cadastral visualisation tool, since it was not developed specifically for that purpose. This paper focuses primarily on 3D visualisation, not on data management or data delivery. The 3D visualisation requirements against which CityEngine was evaluated are classified into three main categories: cadastral requirements, visualisation requirements and non-functional requirements. The evaluation is made through a case study corresponding to a real situation in Portugal previously identified. Results obtained are promising, however it is necessary to carefully study other complex cases. However, the learning curve is steep and the CityEngine will not be the best option for all types of users.

Exploring CityEngine as a Visualisation Tool for 3D Cadastre

Alexandra RIBEIRO, José-Paulo DUARTE de ALMEIDA, Portugal and Claire ELLUL, UK

1. INTRODUCTION

3D cadastre concerns a cadastral system which represents the legal status of a property, not just as a 2D cadastral parcel, but also as a 3D juridical unit with rights, responsibilities and unique and homogenous restrictions (RRRs) (Van Oosterom et al 2011; Stoter, 2004). Development of an effective 3D cadastre system involves consideration of several legal, institutional and technical aspects (Poliout, 2011; Stoter, 2004). Regarding to technical aspects, 3D data acquisition, 3D data modelling and 3D database management systems are considered in the development of 3D cadastral applications. Still, the 3D visualisation also plays a significant role and has come to be recognized as a matter of extreme importance in successive international workshops (1st, 2nd and 3rd International Workshop on 3D Cadastres) by several authors (Poliout, 2011; Fendel, 2001).

3D visualisation enables graphical identification and spatial communication of the complexity of a large number of real life situations of overlapping and encroachments in 2D or 3D land and property interests (e.g., buildings with complex architecture, infrastructures above or below the Earth surface, natural resources and corresponding rights). Visualisation systems are necessary not only to represent physical objects, but also to visualize their legal counterparts. The legal counterparts can either be limited or unlimited volumes (Lemmen et al, 2010).

Several specific prototypes have been proposed for the cadastre, such as spatial databases, with CAD and GIS front-ends (Abdul Rahmanet et al, 2011; Aditya et al, 2009; Billen & Zlatanova, 2003; Hassan et al, 2008; Stoter, 2004; Stoter & Ploeger, 2003; Stoter & van Oosterom, 2006; Ying et al, 2011). On the other hand, the technological advancements and the ease in the use of web-based visualisation applications have made them very popular amongst users. This popularity has lead to the construction of several 3D visualisation prototypes with the use of web technologies (Shojaei et al 2014, 2012; Dimovski et al, 2011; Vandysheva et al, 2012; Elizarova et al, 2012; Ying et al, 2012; Aditya et al, 2011; Guo et al, 2011; Lemmen et al, 2010; Stoter, 2004; Coors, 2003; Stoter & Salzmann, 2003). However, according to Poliout (2011), they still require maturation and validation by the users before being able to be used in real life situations. Thus, research on 3D cadastral visualisation needs more investigation (van Oosterom 2012, 2013; Pouliot 2011).

3D modelling of urban environments utilizes 3D visualisation systems. If these systems can somehow be reutilized in the 3D cadastre context, associate costs might be lower by building a system out of scratch. One of those systems is ESRI CityEngine (http://www.esri.com/software/cityengine). "CityEngine is a three-dimensional (3D) modelling software application developed by Esri R&D Center Zurich (formerly Procedural Inc.) and is specialized in the generation of 3D urban environments. With the procedural

modeling approach, CityEngine enables the efficient creation of detailed large-scale 3D city models with merely a few clicks of the mouse instead of the time-exhaustive and work-intensive method of object creation and manual placement."
(http://en.wikipedia.org/wiki/CityEngine).

This work proposes the evaluation of CityEngine's suitability as a 3D cadastral visualisation tool, as it was not developed specifically for that purpose. The focus of this paper is principally on 3D visualisation, not on data management or data delivery. As a matter of fact, van Oosterom (2013) stated that, "The visualisation and/or interaction with 3D cadastral parcels requires more attention and may be quite different from the more well-known visualisation of 3D city models (Wang et al, 2012)." And added: "Some specific key points are as follows: (1) how to visualize dense 3D volumetric partitions such as in a complex building because the first visible outside layer of 3D spatial units blocks a view of the others; […] (2) how to display open or unbounded parcels, (3) how to include the earth's surface and/or other reference objects (e.g., CityGML-like) for 3D cadastral parcels, (4) how to provide the proper depth cues for subsurface legal spaces related to utilities […]."

Shojaei et al (2013) proposed a group of requisites to be satisfied by 3D cadastral visualisation systems and tested them with several visualisation systems to determine to which extent they fitted the required requisites. Shojaei et al (2014) also reviewed and compared several common 3D web-based visualisation solutions. Google Earth (www.google.com/earth/), a very popular 3D visualisation application, and NASA World Wind (worldwind.arc.nasa.gov/java), were excluded because they both fail to represent underground objects, such as infrastructures or easement rights, which are very important in cadastres. CityEngine was not included in none of those evaluations.

This paper is organised as follows. First, 3D cadastral visualisation requirements are reviewed and the features against which CityEngine is evaluated are identified. Secondly, an overview of CityEngine is conducted (Section 3), and then procedural modelling and Computer Generated Architecture (CGA) shape grammar, two core concepts of CityEngine, are introduced (Section 4). In Sections 6 and 7, a 3D cadastre case study in Portugal and results are presented, respectively. Finally, conclusions and future work are presented.

2. REVIEW OF 3D CADASTRAL VISUALISATION REQUIREMENTS

Shojaei et al (2013) identified a comprehensive set of requirements for 3D cadastral visualisation, based on a review of the literature and also through a consultative workshop with industry partners. These requirements were classified into three main categories: cadastral requirements, visualisation requirements and non-functional requirements (Table 1).
Cadastral requirements include the essential elements in developing efficient and effective cadastral applications to represent 3D properties. Visualisation requirements are a set of features that are widely used in general 3D visualisation applications to facilitate communication with end users. Non-functional requirements provide support for technical diversity, system interoperability and integration and usability.

In this paper, the evaluation of CityEngine’s potential was performed against all the aspects in Table 1, except handling massive data.

Table 1. The list of 3D cadastral visualisation requirements

Features	Visualisation requirements	Description
Cadastral features	Handling massive data	Representing massive cadastral data using visualisation techniques
	Result of functions and queries	Visualising results of cadastral functions and queries
	Underground view	Representing objects beneath ground level
	Cross-section view	Slicing an object at a plane
	Measurements (3D)	Measuring unofficial distances or areas
	Display non-spatial data	Illustrating legal documents attached to each development
Visualisation features	Interactivity	Required tools for exploring a 3D scene
	Levels of detail	Visualisation technique for accelerating the rendering process
	Symbols	Cartographical elements
	Colour, thickness, line- style	Object properties for visualisation of data
	Labelling	Annotations attached to objects on a scene
	Transparency	Object properties for visualisation of data
	Tooltips	An identify tool to presents attribute data
Non-functional features	Technical diversity	Diversity in supported technology
	System integration and interoperability	The ability to exchange data and connect different components of applications
	Usability	Ease of use and learnability
	Platform independence	Independence from a specific platform
	Cost	Cost of developing and maintenance a visualisation application
	Web-based 3D visualisation	Web-based solution

Source: Shojaei et al (2013)

3. ESRI CityEngine OVERVIEW

CityEngine is a stand-alone desktop commercial application for the design, planning, and modelling of urban environments in 3D (Figure 1). It was created to facilitate professional users in GIS, CAD, and 3D to: i) quickly generate 3D cities from existing 2D GIS data; ii) do conceptual design in 3D, based on GIS data and procedural rules; and iii) model virtual 3D urban environments for simulation and entertainment. Consequently, professionals in the following industries use CityEngine: urban planning, design and development (architectural visualisation and local government); entertainment (films, commercials, videogames); real world simulation, emergency response and defense; and, of course, the Academia.

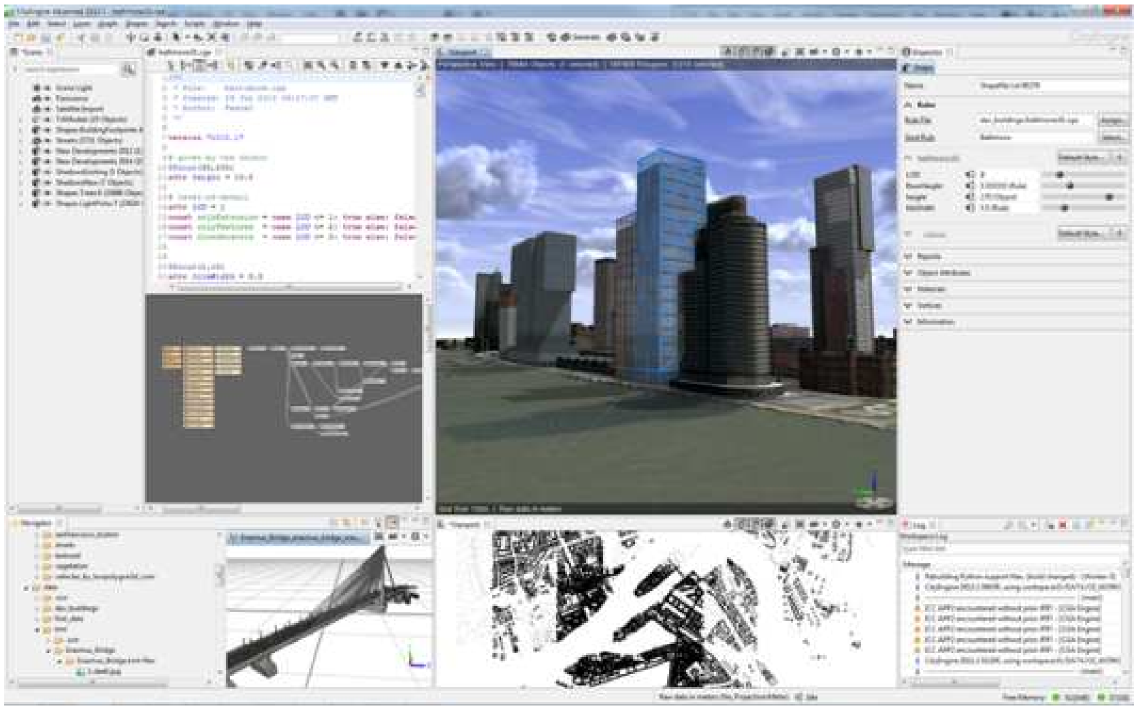


Figure 1. CityEngine main window (Source: CityEngine help)

It runs on all three major operating systems. A native 64-bit version is available for Windows (7/8/8.1), Linux and Mac OS X with Intel processors. The current release (2014.1) connects to the rest of the ArcGIS system primarily through data exchange. CityEngine imports/exports several file formats (Figure 2). Terrain models can be created from simple image files or from Digital Elevation Models. In the latter case – for example, with a GeoTIFF file – georeferencing information is supported. Currently, CityEngine only supports image-based terrains (gray scale height maps); it does not support 3D meshes. CityEngine provides tools to align shapes to the terrain.

The main concept of the CityEngine is the “procedural” approach towards efficiently modelling. The computer is given a code based “procedure” which represents a series of commands - in this context, geometric modelling commands - which then will be executed.

Instead of the “classical” intervention of the user, who manually interacts with the model and models 3D geometries, the task is described “abstractly”, in a rule file. The commands which are provided in the CityEngine's CGA shape grammar, such as “extrude”, “split” or “texture”, are widely known commands in most 3D applications and thus any user can adapt them easily and create complex architectural forms in relatively short time.

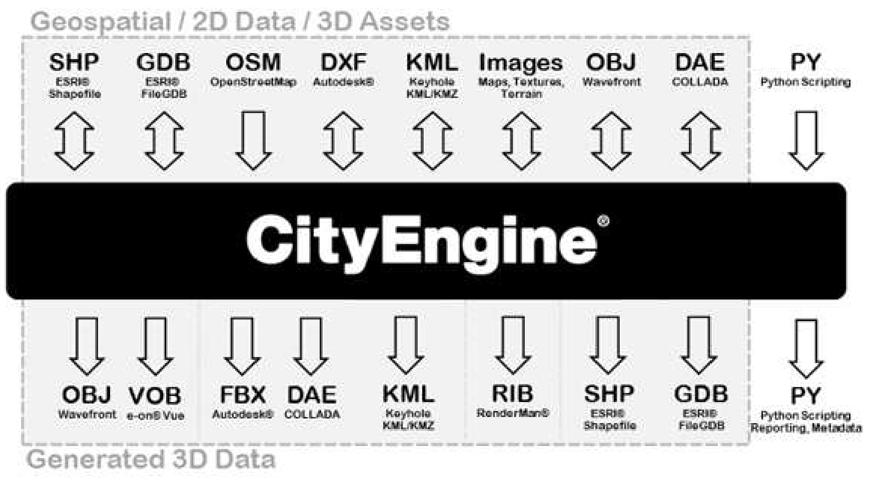


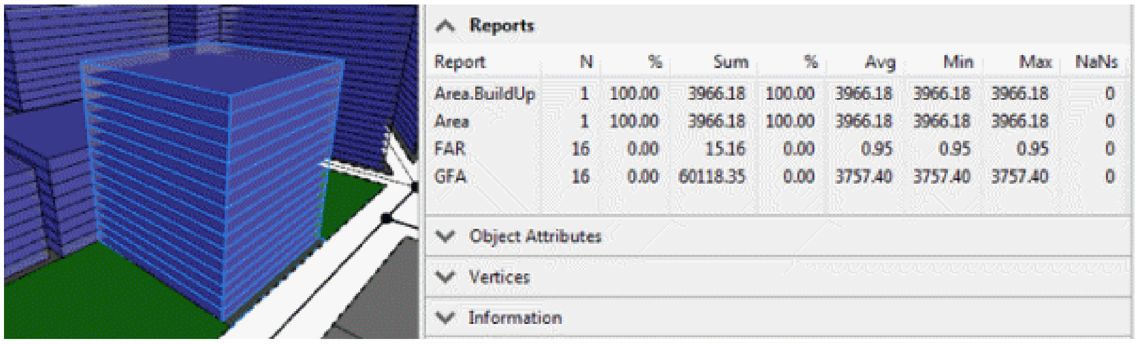
Figure 2. CityEngine import/export file formats (Source: CityEngine help)

A single procedural rule can be used to generate many 3D models. For example, the rule can make use of feature attribute information stored in GIS data – such as, the number of floors, floor height, roof type, wall material type, etc. – to generate a series of alternate 3D models that accurately represent properties of each feature. The more attributes you have, the more accurate the generated model may be. A 3D model is no more than a 3D object resulting from a 2D shape extrusion according to the rules defined in a CGA rule file. The origin of these 2D shapes is variable: i) they can be imported from ESRI Shapefiles or File Geodatabases: ii) be built manually in CityEngine; iii) or, also be generated through CGA rules.

The three-dimensional objects represented in the CityEngine don't have all of them to be generated within CityEngine. They can be imported through the aforementioned formats (Figure 2). However, only the geometry of objects from multipatch ESRI Shapefiles or File Geodatabases can be edited and later upgraded in source files. 3D models designate the 3D objects generated in CityEngine through procedural modelling. The remaining objects (3D or 2D) are referred to as Shapes.

Customized rule-based reports can be created. The report operation allows for the reporting of arbitrary properties of one or a set of selected objects (Figure 3). Consequently, the reporting procedure is completely generic and customizable. For example, it can include numbers, such as gross floor areas (GFA), volume, number of units, or land use mixes. In addition, by changing the urban design (that is, regenerating the models), reports are updated automatically and instantaneously.

Advanced Edition comes with an integrated Python scripting interface. Through Python scripting users can have control over repetitive tasks, create formatted reports in file format, or automate other specific actions. Attribute queries are also possible, but only through Python scripting. Any commands accessible from the graphical interface of the CityEngine is available in Python scripting interface.



Report	N	%	Sum	%	Avg	Min	Max	NaNs
Area.BuildUp	1	100.00	3966.18	100.00	3966.18	3966.18	3966.18	0
Area	1	100.00	3966.18	100.00	3966.18	3966.18	3966.18	0
FAR	16	0.00	15.16	0.00	0.95	0.95	0.95	0
GFA	16	0.00	60118.35	0.00	3757.40	3757.40	3757.40	0

Figure 3. Reports pane in the Inspector. Values for a selected building. (Source: CityEngine Help)

CityEngine is not a web client. It does not contain any web-enabled capabilities, though it can export generated content out to ArcGIS Online. A 3D CityEngine Web Scene is a 3D scene with limited extent. It can be hosted on ArcGIS Online and viewed in 3D in the browser. The recommended maximum unzipped size is 70 MB (CityEngine help). The 3D Web Scene Viewer is an application on ArcGIS Online that allows you to view 3D Web Scenes. No plugin is required for most browsers. Layer views are supported and the swipe tool can be used for a side-by-side view of visible layers. Dynamic shadows, bookmarks and a search tool are also available in the 3D Web Scene Viewer. The 3D Web Scene is an export format (WebGL) of CityEngine. Any 3D format that can be imported into CityEngine can be exported to a 3D Web Scene. Because the 3D Web Scene Viewer is a web application for interacting with 3D images, one needs to use a web browser that supports WebGL.

4. PROCEDURAL MODELLING AND CGA SHAPE GRAMMAR

In this section the concepts of procedural modelling with the CGA shape grammar follow closely the ones explained in CityEngine help. The CGA shape grammar of the CityEngine is a programming language indicated to generate architectural 3D content (Müller et al, 2006). The idea of grammar-based modelling is to define rules that iteratively refine a design by creating more and more detail.

The following rule derivation illustrates the process: on the left side the initial shape is shown and on the right side the resulting generated model is displayed (Figure 4).

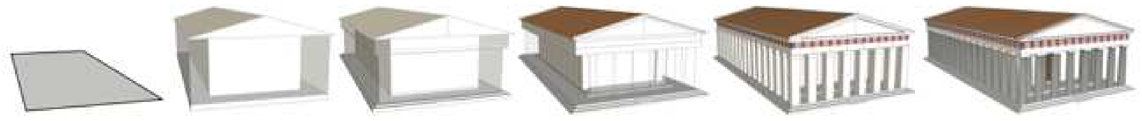


Figure 4. Illustration of the process of defining rules that iteratively refine a design by creating more and more detail. (Source: CityEngine help)

These rules operate on shapes. In short, a shape consists of a name (so-called shape symbol or rule name); a geometry; a locally oriented bounding-box (so-called scope) in space relative to the pivot; and, a pivot, describing the shape's coordinate system. The geometry of a shape is a polygonal mesh, and attributes like colour, material and textures are also included in the geometry. The pivot is given in object coordinates, relative to the initial shape's origin.
The elementary idea of a rule is to replace a shape with a certain shape symbol with a number of new shapes. Formally:

```
PredecessorShape --> Successor
```

In this very simple rule, `A --> B,` the rule creates a copy of the shape A and sets its shape symbol to B. The A shape is now considered done and not processed anymore. If there is no rule matching symbol B the generation process is finished. The resulting structure is called the *shape tree* (see Figure 5, on the left).
In the extremely simple shape tree above, A is the root shape and B is a *leaf shape*. Leaves are very important because the sum of all leaves represents the generated model Inner nodes are not visible in the final model (see Figure 5, on the right).

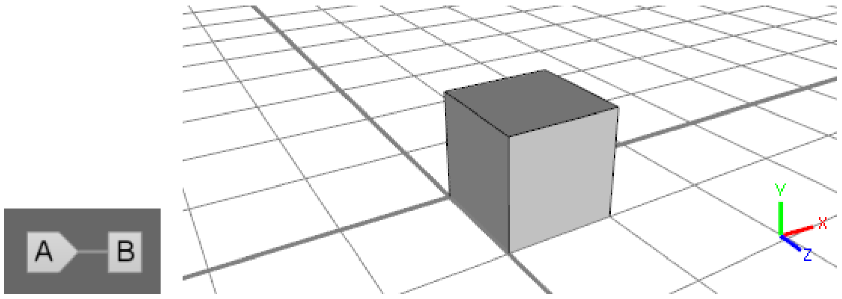


Figure 5. Shape tree (on the left) and final model of the generation process (on the right). (Source: CityEngine help)

In this very simple example, it is assumed that shape A's geometry, scope and pivot are set up such that the shape represents a unit cube in the origin; because B is a copy of A, B looks exactly the same (see the picture above). A rule can have more complex successors, e.g., the right side of the rule can consist of multiple shape symbols and so-called shape operations:

```
A --> B t(3, 0, 0) C
```

This successor is now executed from the left to the right. B is an identical copy of A. Then, the current shape is translated by 3 units in x direction (*i.e.*, the scope.t is manipulated), and a new shape C is created. The shape tree now has two leaves, B and C. Adding these two rules:

```
C --> D s(2, 0.5, 1.75) E
E --> i("cylinder.obj") F
```

the generation process will add two children, D and E to shape C. Shape D is an exact copy of shape C, but shape E will have a different sized scope (because of the s() shape operation). After the shape insert operation i(), shape E is not a leaf anymore but now has a child shape F. The geometry of shape F is not a cube anymore but was replaced with the mesh read from file "cylinder.obj". Note that the size (i.e. the scope.s vector) of shape F is still the same as the one of shape E. The leaves (B, D, F) are not on the same level (*i.e.* have different distances to the root shape) but they are all part of the 3D model (Figure 6).

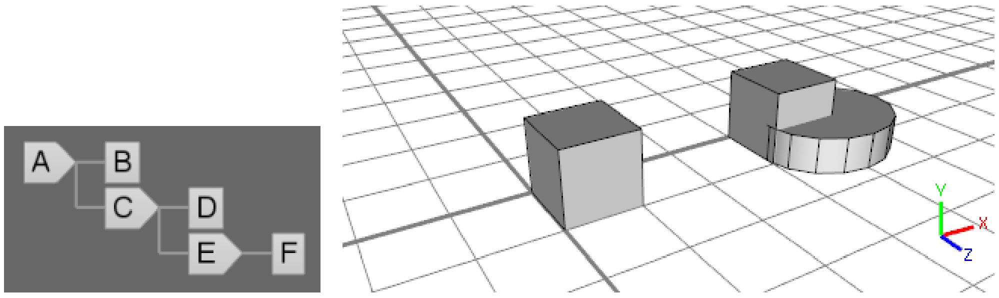
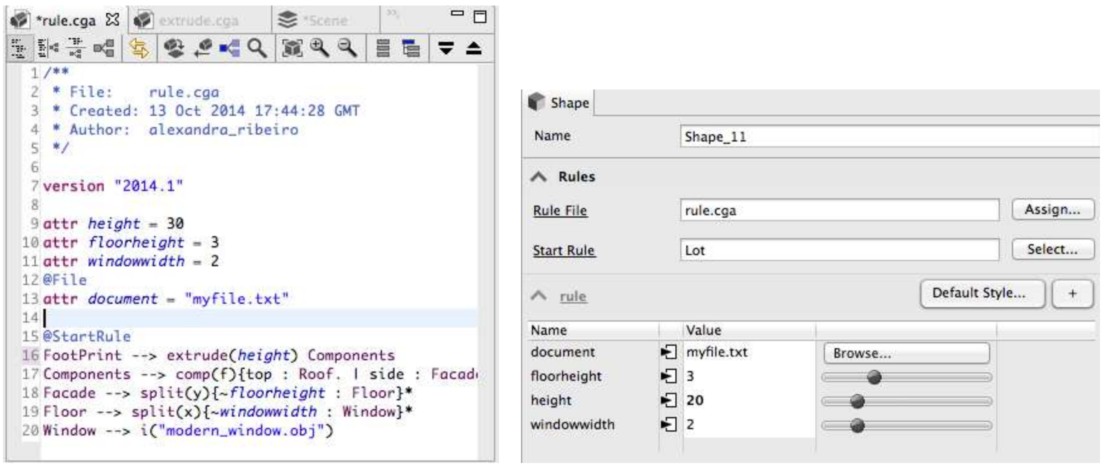


Figure 6. The shape tree (on the left) and the associated model (on the right). (Source: CityEngine help)

Rules are saved in a rule file and created in the CityEngine's CGA editor. In Figure 7 (on the left) is shown an example of a rule file. *height*, *floorheight*, *windowwidth* and *document* (prefixed by *attr*) are attributes automatically displayed in the Inspector Window (Figure 7, on the right). CityEngine generates the sliders and the browse button. The user can modify the attributes' values in the Inspector Window for each shape. These attributes can be tied up to GIS data attributes (*e.g.*, fields of a Shapefile's attribute table) and the correspondent values updated back. It should be noted that even the specification of the colour or transparency of an object passes through the definition of a rule.



```
/**
 * File:    rule.cga
 * Created: 13 Oct 2014 17:44:28 GMT
 * Author:  alexandra_ribeiro
 */

version "2014.1"

attr height = 30
attr floorheight = 3
attr windowwidth = 2
@File
attr document = "myfile.txt"

@StartRule
FootPrint --> extrude(height) Components
Components --> comp(f){top : Roof. | side : Facad
Facade --> split(y){~floorheight : Floor}*
Floor --> split(x){~windowwidth : Window}*
Window --> i("modern_window.obj")
```

Name	Shape_11
Rule File	rule.cga
Start Rule	Lot

Name	Value
document	myfile.txt
floorheight	3
height	20
windowwidth	2

Figure 7. Example of a CGA rule file (on the left) and the corresponding Inspector Window (on the right)

In the next step, the newly created rule file has to be assigned to the corresponding shapes (in the above case to building footprints). These forms must be previously selected and may be just one or more, and can be distributed across several layers.

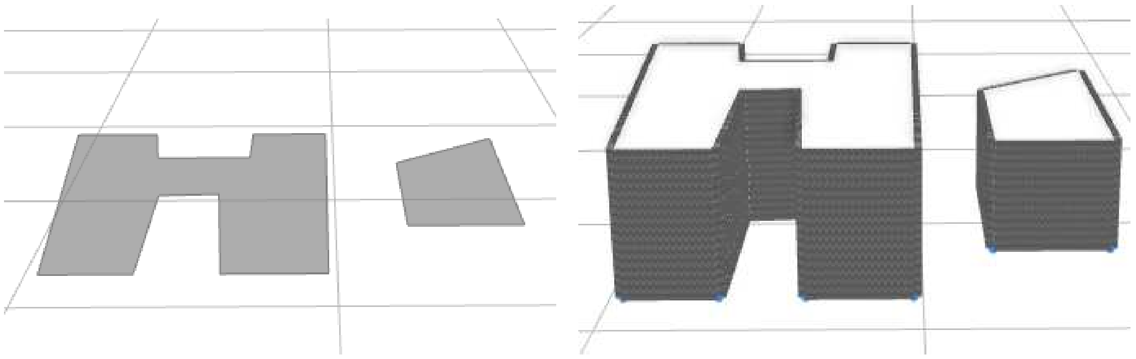


Figure 8. Two shapes, e.g., building footprints (on the left), that was applied the CGA rule above and the corresponding generated 3D models (on the right)

5. THE CityEngine MODELLING PIPELINE AND POSSIBLE ADAPTATION FOR THE CASE OF 3D CADASTRAL VISUALISATION

Modelling an urban environment with the CityEngine usually implies the individual stages of the pipeline given in Figure 9. As the main idea behind the CityEngine is generating an urban environment from scratch, the first stage consists in the generation of street centrelines based on graph algorithms pre-programmed, and whose values of the parameters the user can modify. Then the street lines give rise to 2D polygons representing the area occupied the streets, and lots (2D parcels) are created. In both cases, the user can modify the values of the parameters used. In a third stage, the buildings and are built in 3D, based on CGA rules files, given the number of floors, each floor height, type of roof, etc., including colour and texture. The 3D buildings can be generated directly from the lots, specifying, *e.g.*, setbacks, or from previously imported 2D footprints. In this stage, vegetation, water bodies, vehicles, street furniture, etc., can be also generated based on CGA rules and 3D assets. Infrastructures above and below Earth surface are also represented in this stage. Finally, the urban environment obtained (Figure 10) can be exported to other software specialized in 3D visualization, presentation or analysis (*e.g.*, ArcGIS, Maya, 3ds Max, Google Earth, Unity, Unreal, RenderMan, and RealityServer).

This pipeline must be adapted for the case of 3D cadastral visualisation. Streets and lots do not have to be generated accordingly with some user-defined or default values for parameters. They should be imported from the cadastral system through ESRI Shapefiles (multipatch or not) or File Geodatabases (multipatch or not). This means that data should be first converted to these file formats. Extract, Transform and Load (ETL) tools can be helpful in these situations, or, some code should be developed instead. Not only feature geometry is imported, but also feature attribute information, which is displayed in the Inspector Window.

For imported 3D objects may be only necessary to modify the way they are viewed (colour, thickness, transparency, texture, etc.). In the case of 2D objects, must be additionally defined extrusion CGA rules based on, for instance, attributes from GIS attribute tables.

Another important CityEngine's characteristic is the fact that, in the case of features from a File Geodatabase, existing relations between tables are imported (only in version 2014.1); they can be viewed and explored through code (CGA rules or Python). Suppose the flat A, situated on the level L of a building B located on the parcel P, is the property of the owners O1 and O2. If these relations are modelled in the File Geodatabase, then in the Inspector Window, you will see that information regarding the physical 3D object "Flat A". The attribute "Owner" of the object "Flat A" will appear in the form of list, since there are two owners – O1 and O2.

Programming in Python can also introduce a great deal of flexibility to the workflow. By programming in Python it is possible to define which objects must belong to every layer of a scene (set of layers). For example, there may be interest in separate legal objects in the public domain of the private domain, placing them in different layers, and later, separately control their visibility. Or, in the case of a block of flats in particular, separating the flats per floor and set a layer for each floor. Moreover, attribute queries are only feasible through programming in Python. For example, select all objects (flats) with more than one owner, and then create a layer with this information; or export this information to a text file.

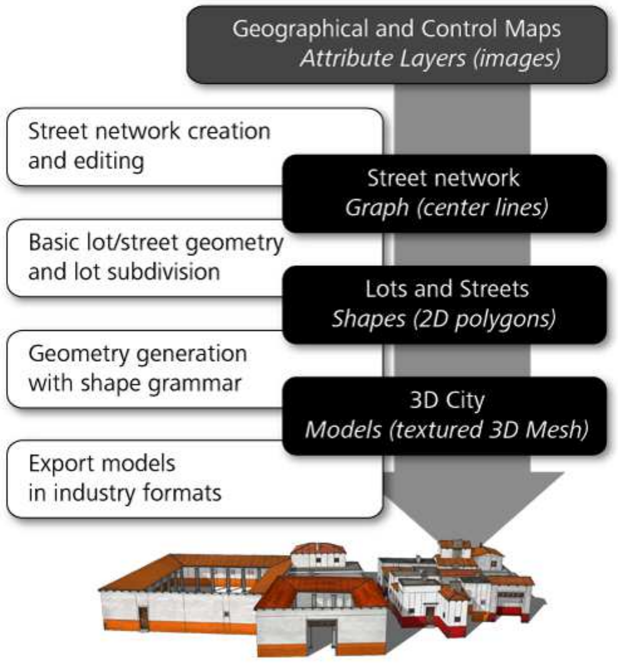

Geographical and Control Maps
Attribute Layers (images)
Street network creation and editing
Street network
Graph (center lines)
Basic lot/street geometry and lot subdivision
Lots and Streets
Shapes (2D polygons)
Geometry generation with shape grammar
3D City
Models (textured 3D Mesh)
Export models in industry formats

Figure 9. Overview of the CityEngine typical modelling pipeline. Black boxes illustrate data types (layers) and white boxes the operations to create them. Typically, in the first step, the street network is created; then resulting blocks are subdivided into lots afterwards. Finally, the 3D models of the buildings are generated using the CGA rules. The main output of CityEngine are polygonal building models. (Source: CityEngine help)

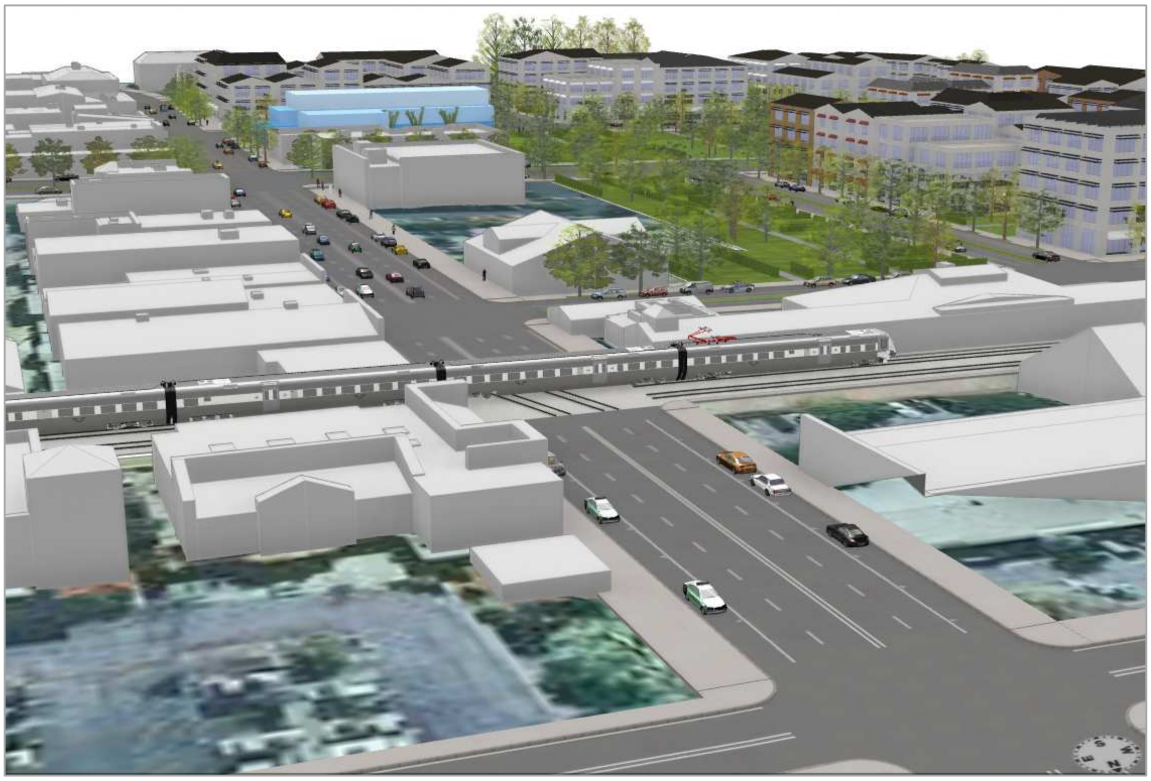


Figure 10. Example of a 3D city generated by CityEngine through procedural modelling

6. CASE STUDY

The most recent initiative of the Portuguese government towards the implementation of a centralised cadastral management system is the design and implementation of the so-called SiNErGIC (PCM 2006). The main drive of SiNErGIC is to accomplish a multipurpose cadastral system in Portugal setup as an "exhaustive, methodical, and up-to-date set of data able to uniquely identify and describe property parcels" (DGT 2012). Cadastral surveying is currently being accomplished district-by-district covering both kinds of properties, rural and urban. By the end of 2011 more than 50% of the mainland was surveyed, however this represents roughly 1/3 of the total number of properties in the country (de Almeida et al, 2014). Those surveys covered mostly rural properties. Currently, 7 districts are being surveyed in Portugal's mainland.

There are some concepts of cadastral spaces specific of the Portuguese cadastral law (de Almeida et al, 2014), which need be introduced prior to the case study. Those concepts are:

- The "municipal domain" (in the Portuguese legislation, "domínio privado municipal"), which stands for state rights over a particular real estate – land parcel or manmade infrastructure – owned by the local city/town council whose jurisdiction covers the district territory where the given property happens to be located;

- The “public domain” (in the Portuguese legislation, “domínio público”) stands for citizenship rights over the general public space – managed though by a specific state agency, depending on each instance;
- The “private domain” (in the Portuguese legislation, “domínio privado particular”) stands for private rights over a particular real estate – land parcel or manmade infrastructure – owned by a single or any sort of corporate person.

Many examples can be recognised in Portugal where the 2D cadastre is not sufficient. 3D aspects of cadastral data have not been covered in SiNErGIC. One of those examples is presented here.
The case study relates to a building of private flats over an urban road (Figure 11), located at a Portuguese small town – Figueira da Foz. The building has two arches over the road. This represents an exception to the legal principle in the Portuguese cadastral law – ownership rights over a given real estate on the ground also apply to both areas above and underneath that property. Definitely, condominium ownership rights cannot be applied to the over ground area underneath the arches since this represents public domain.

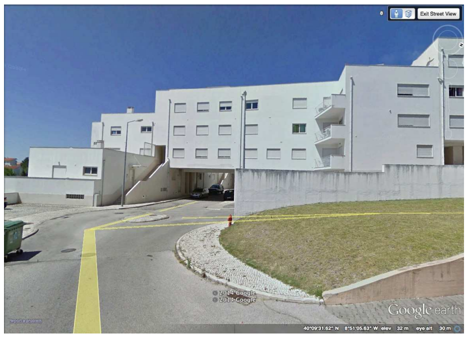
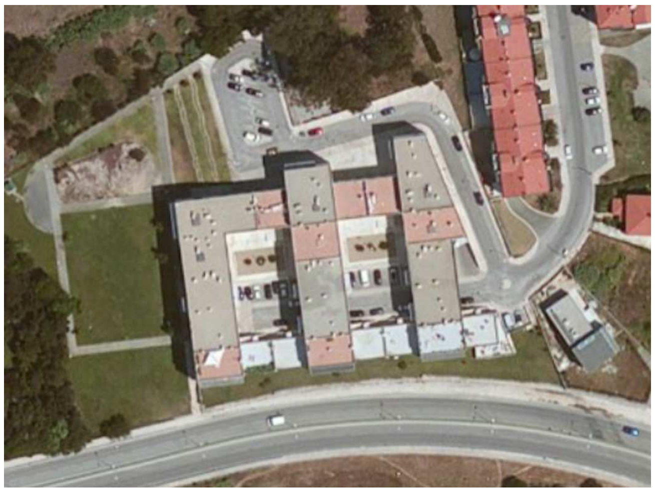
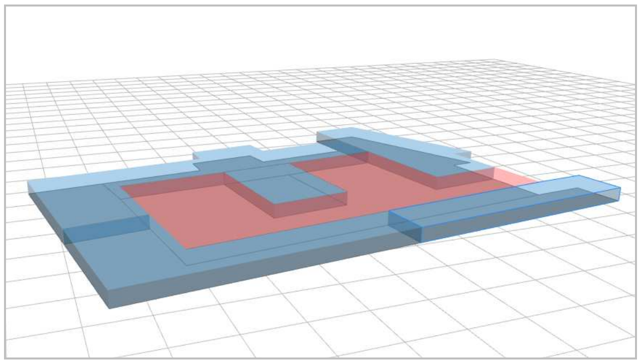



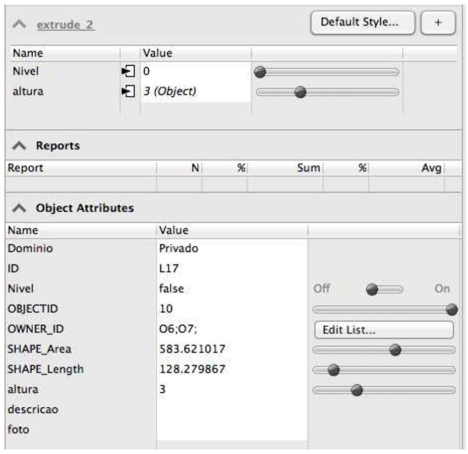
Figure 11. A two arch-building of private flats (private domain) over an urban road (public domain). (**Source: Google Earth)**

Since the only information known about the building comes down to photographs and a 2D map taken from Google Earth, only a schematic representation was created. The plants of each floor were outlined on the 2D map, in ESRI ArcMap, and saved as a feature class in a File Geodatabase. A polygon feature in that feature class represents each flat or the area corresponding to the public domain. The distribution of the flats per floor is completely fictitious. Every feature class (floor) were added several fields, *e.g.*, the number of the floor, the floor height, the entity ID. It was also created an alphanumeric table with three columns, one containing the entity ID (regardless of the floor where they are), the second the owner ID and the third the type of domain (private or public). In this table it was considered that some flats have more than one owner.

In the schema, two floors above the ground and one floor below were contemplated. It was considered a height of 3 m to the floors above the surface and 4 m to the floor of the basement.

The File Geodatabase was imported to the CityEngine. The 2D objects of each floor were placed in the same layer. It was created a CGA rule file with all the necessary rules to: i) make the object extrusion; ii) giving colour depending on the type of cadastral domain (red colour for the public domain and blue colour for the private domain); iii) provide transparency. These rules were applied simultaneously to all objects.
Controlling the visibility of layers, it is possible to see a floor at a time. It is also possible to show only the selected objects. Whenever an object is selected, its properties are displayed in the Inspector Window (). The selected apartment has two owners (O6 and O7), it is on the first floor ("Nível 0") and it is private domain ("Domínio Privado"). The remaining figures relate to other examples of visualization.





Name	Value
Nivel	0
altura	*3 (Object)*

Reports

Report	N	%	Sum	%	Avg

Object Attributes

Name	Value
Dominio	Privado
ID	L17
Nivel	false
OBJECTID	10
OWNER_ID	O6;O7;
SHAPE_Area	583.621017
SHAPE_Length	128.279867
altura	3
descricao	
foto	

Figure 12. Visualization of 1st floor of the building (on the left) and selected object attributes (on the right); private and public domain objects in blue and red, respectively

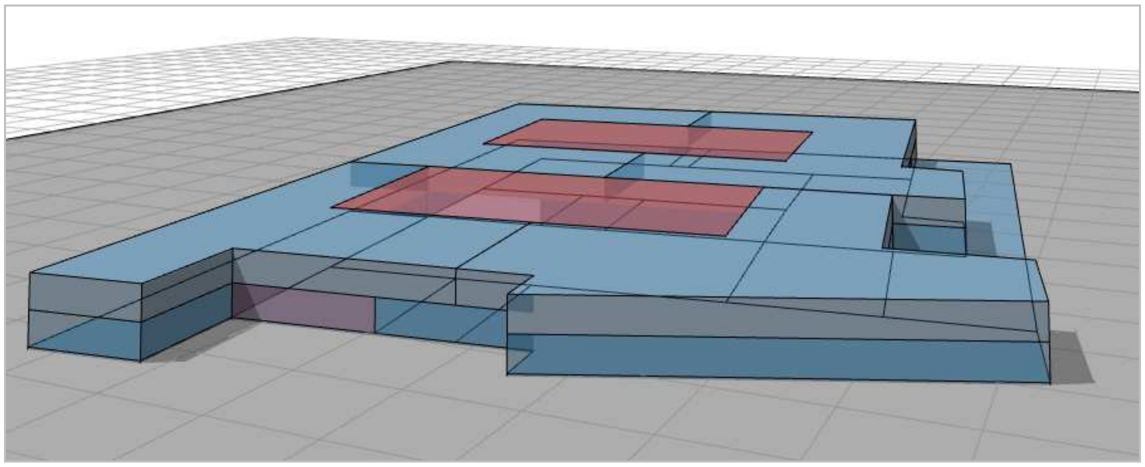


Figure 13. Visualization of 1st and 2nd floors of the building (on the left); transparency, shadows and wireframe applied

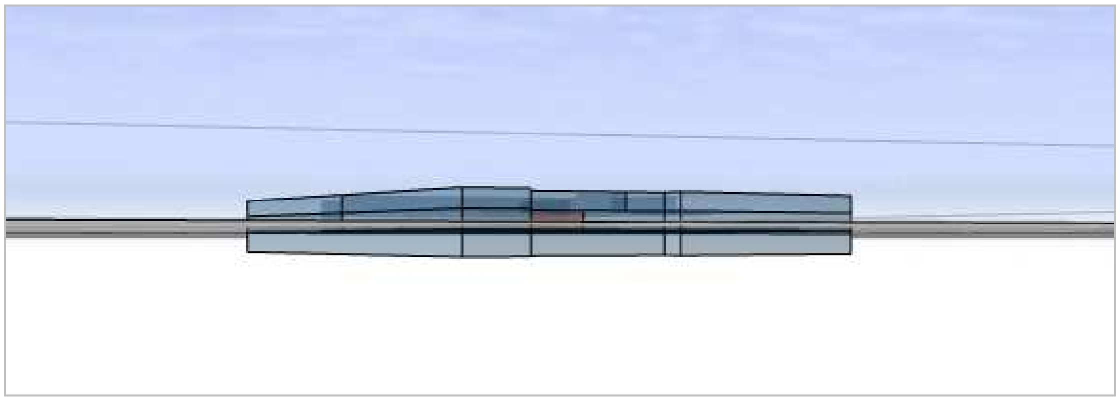


Figure 14. Visualization of all building floors: two of them above de ground and one below

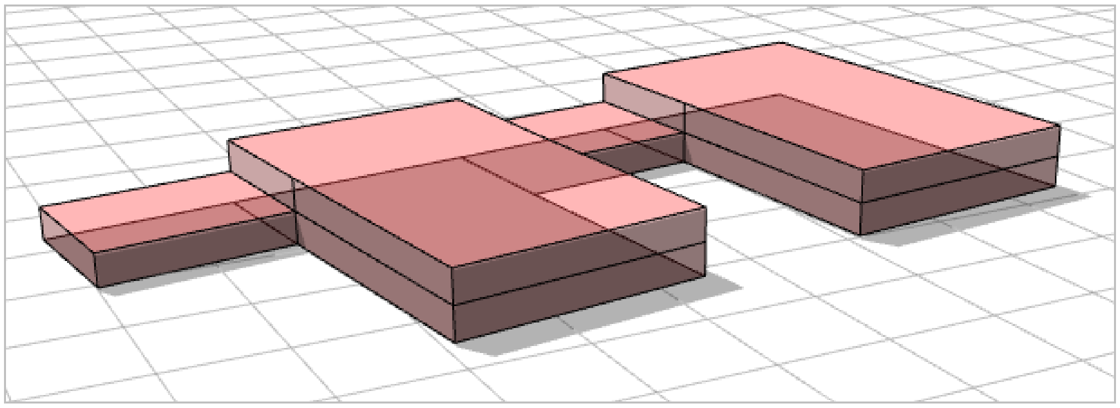


Figure 15. Visualization of objects of public domain

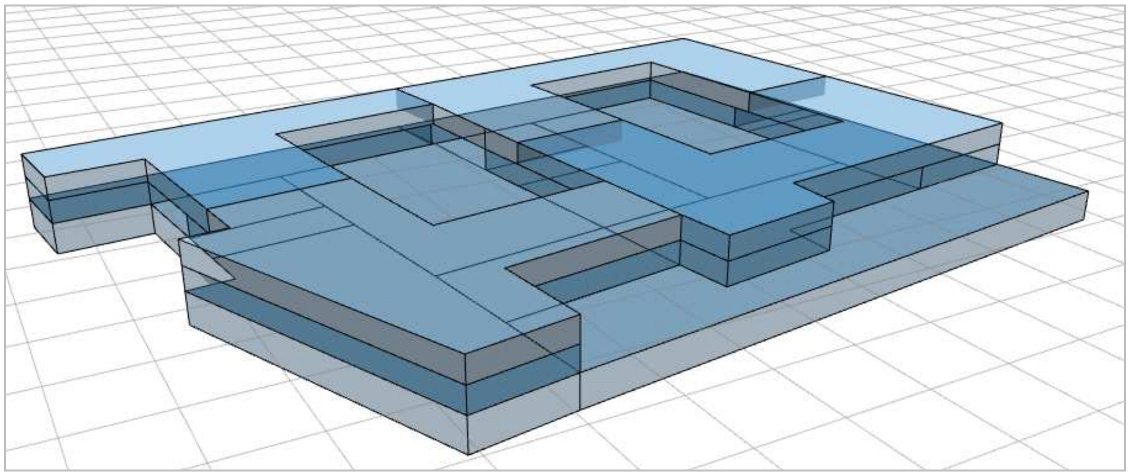


Figure 16. Visualization of objects of private domain

A 3D legal space not unbounded, wholly or partially, may be represented by one or more surfaces of the 3D object slightly away. As each volume can be decomposed into faces, edges and nodes, simply apply two rules: one for the volume decomposition in faces and one for the translation of all or some of them.

7. RESULTS

The experiences made so far have led to the results summarized in Table 2.

Table 2. Results of City Engine's evaluation concerning 3D cadastral visualisation requirements

Features	Visualisation requirements	Evaluation
Cadastral features	Handling massive data	Not tested, although the capabilities of mass modelling foresee the possibility of handling massive cadastral data
	Result of functions and queries	Yes, but using programming
	Underground view	Yes
	Cross-section view	Yes
	Measurements (3D)	Yes, but using programming; there is no tool for direct measurements on a layer
	Display non-spatial data	Yes
Visualisation features	Interactivity	Yes; it is possible to set different viewing perspectives, zooming, set the scene light, view/hide shadows, view/hide textures
	Levels of detail	Yes; schematic visualization of parcels, buildings, streets, etc., till textures of objects, visualization of the buildings' interior, vegetation, street furniture, etc.
	Symbols	No
	Colour, thickness, line-style	Yes; any editable object or 3D model can be decomposed into faces (only for 3D objects), edges and nodes, whose display properties can be controlled
	Labelling	No; but possible to show measurements of objects in the x, y, and z directions through handles
	Transparency	Yes; wire framing is also supported
	Tooltips	No; the properties of any selected object are displayed in a window (the Inspector Window), including the coordinates and attributes
Non-functional features	Technical diversity	Yes; imports and exports the most usual 2D/3D formats, including GIS data.
	System integration and interoperability	Yes
	Usability	Moderated; easy-to-use graphical interface but the need to learn two scripting languages (CGA rules and Python) decreases usability
	Platform independence	Windows, Mac OS X and Linux (Java based)
	Cost	Proprietary solution; at the beginning the learning curve is steep, but the flexibility introduced by the procedural modelling and Python scripting facilitates maintenance – it is a matter of re-apply the same CGA rules and Python code to another dataset
	Web-based 3D visualisation	No; possible to export to a 3D Web Scene

Additionally, CityEngine is able to display aerial and satellite images, and digital elevation models. Connection to web services such as WFS and WMS is not possible. CityEngine is not a 3D analysis tool. Consequently, it is not possible to do shadow analysis nor line of sight and visibility analysis. In those cases, the 3D content created in CityEngine have to be loaded into ArcGIS 3D Analyst for visualization, editing, and analysis. Nevertheless, 3D updating and manipulation is possible to some extent: in the case of GIS data from ESRI Shapefiles and File Geodatabases, there is no problem, but in the other cases, only rotate, scale and shift are allowed.

8. CONCLUSION AND FUTURE WORK

ESRI's CityEngine is a modelling application that allows users to accurately model 3D urban environments for simulation. Its primary strength is in 3D content creation, which uses rule-based modelling to generate detailed 3D objects from 2D data. Rules can make use of attributes stored in GIS data, such as building height and information specific of cadastral systems. The use of rule-based modelling provides an efficient way to construct 3D geometries and textures, rather than relying on labour-intensive manual modelling.

In this work, the CityEngine's suitability as a 3D cadastral visualisation tool was evaluated, as it was not developed specifically for that purpose. The focus of this paper was principally on 3D visualisation, not on data management or data delivery. CityEngine's potential was performed against some previously identified 3D cadastral visualisation requirements. Massive data handling was not tested.

A case study in Portugal was used in the assessment. Results obtained are promising, however it is necessary to carefully study other complex cases. As visualisation systems are necessary not only to represent physical objects, but also to visualize their legal counterparts, this has been considered in the example studied.

The combination of programming in Python and with CGA shape grammar is powerful and gives great flexibility. This flexibility facilitates maintenance – it is a matter of re-applying the same CGA rules and Python code to another dataset. However, the learning curve is steep and the CityEngine will not be the best option for all types of users.

REFERENCES

Abdul Rahman, A., Hua, T. C., & van Oosterom, P. (2011). Embedding 3D into multipurpose cadastre. FIG working week 2011, Marrakech, Morocco.

Aditya, T., F. Iswanto, A. Wirawan, and D. P. Laksono (2011). 3D Cadastre Web Map: Prospects and Developments. In 2nd International Workshop on 3D Cadastres, edited by Fendel, E., Oosterom P. van, H. Ploeger, J. Stoter, A. Streilein, and T. Tijssen, November 16–18. Delft, the Netherlands.

Aditya, T., Subaryono, Waljiyanto, Istarno, Diyono, & Untung, R. (2009). Understanding the urgency for 3D cadastre in Indonesia: Development and visualization of a hybrid 3D cadastre model. In 10th South East Asian survey congress, Bali, Indonesia.

Billen, R., & Zlatanova, S. (2003). 3D spatial relationships model: A useful concept for 3D cadastre? Computers, Environment and Urban Systems, 27(4), 411–425.

Coors, V. (2003). 3D-GIS in Networking Environments. Computers, Environment and Urban Systems 27 (4): 345–357. doi:10.1016/S0198-9715(02)00035-2.

de Almeida, J.-P., Liu, X., Ellul, C., & Rodrigues-de-Carvalho, M. M. (2014). Towards a Property Registry 3D Model in Portugal: Preliminary Case Study Implementation Tests. In U. Isikdag (Ed.), Innovations in 3D Geo-Information Sciences (pp. 291–320). Cham: Springer International Publishing. Retrieved from http://link.springer.com/10.1007/978-3-319-00515-7_17.

Dimovski, V., Bundaleska-Pecalevska, M., Cubrinoski, A., & Lazoroska, T. (2011). WEB portal for dissemination of spatial data and services for the needs of the agency for real estate cadastre of the Republic of Macedonia (AREC). In 2nd International Workshop on 3D cadastres, Delft, the Netherlands.

DGT (2012). Instituto Geográfico Português. www.igeo.pt. Accessed 10 Oct. 2014.

Elizarova, G., S. Sapelnikov, N. Vandysheva, S. Pakhomov, P. V. Oosterom, M. D. Vries, J. Stoter, et al (2012). Russian-Dutch Project '3D Cadastre Modelling in Russia.' 3rd International Workshop on 3D Cadastres: Developments and Practices, Shenzhen, China.

Federation. FIG Working Week 2012, Rome, Italy.

Guo, R., L. Li, B. He, P. Luo, S. Ying, Z. Zhao, and R. Jiang (2011). 3D Cadastre in China – A Case Study in Shenzhen City. 2nd International Workshop on 3D Cadastres, Delft, The Netherlands.

Hassan, M. I., Ahmad-Nasruddin, M. H., Yaakop, I. A., & Abdul Rahman, A. (2008). An integrated 3D cadastre – Malaysia as an example. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences.

Lemmen, C., van Oosterom, P., Thompson, R., Hespanha, J., & Uitermark, H. (2010). The modelling of spatial units (parcels) in the land administration domain model (LADM). FIG congress 2010, Sydney, Australia.

Müller, P., Wonka, P., Haegler, S., Ulmer, A., & Van Gool, L. (2006). Procedural Modeling of Buildings. In ACM SIGGRAPH 2006 Papers (pp. 614–623). New York, NY, USA: ACM. doi:10.1145/1179352.1141931.

PCM-Presidência do Conselho de Ministros (2006). Resolução do Conselho de Ministros nr. 45. Diário da República Portuguesa, Série I-B, No. 86, 4 de maio.

Pouliot, J. (2011). Visualization, distribution and delivery of 3D parcels. In 2nd International workshop on 3D Cadastres, Delft, The Netherlands.

Shojaei, D., M. Kalantari, I. D. Bishop, A. Rajabifard, and A. Aien (2013). Visualization Requirements for 3D Cadastral Systems. Computers, Environment and Urban Systems 41: 39–54. doi:10.1016/j.compenvurbsys.2013.04.003.

Shojaei, D., Rajabifard, A., Kalantari, M., Bishop, I. D., & Aien, A. (2012). Development of a 3D ePlan/LandXML visualisation system in Australia. In 3rd International workshop on 3D cadastres: Developments and Practices, Shenzhen, China.

Stoter, J. E. (2004). 3D cadastre. PhD Thesis, TU Delft, Delft, the Netherlands.

Stoter, J. E., & Ploeger, H. D. (2003). Property in 3D – Registration of multiple use of space: Current practice in Holland and the need for a 3D cadastre. Computers, Environment and Urban Systems, 27, 553–570.

Stoter, J., and M. Salzmann (2003). Towards a 3D Cadastre: Where Do Cadastral Needs and Technical Possibilities Meet? Computers, Environment and Urban Systems, Environment and Urban Systems 27 (4): 395–410. doi:10.1016/S0198-9715(02)00039-X.

Van Oosterom, P. (2012). Summary of the Third International FIG Workshop on 3D Cadastres – Developments and Practices. 3rd International Workshop on 3D Cadastres – Developments and ractices, Shenzhen, China.

Van Oosterom, P. (2013). Research and Development in 3D Cadastres. Computers, Environment and Urban Systems 40: 1–6. doi:10.1016/j.compenvurbsys.2013.01.002.

Van Oosterom, P., Stoter, J., Ploeger, H., Thompson, R., & Karki, S. (2011). World-wideinventory of the status of 3D cadastres in 2010 and expectations for 2014. FIG working week 2011, Marrakech, Morocco.

Vandysheva, N., S. Sapelnikov, P. van Oosterom, M. de Vries, B. Spiering, R. Wouters, A. Hogeveen, and V. Penkov (2012). The 3D Cadastre Prototype and Pilot in the Russian Republic.

Ying, S., Guo, R., Li, L., van Oosterom, P., Ledoux, H., & Stoter, J. E. (2011). Design and development of a 3D cadastral system prototype based on the LADM and 3D topology. In 2nd International Workshop on 3D Cadastres, Delft, the Netherlands.

Ying, S., R. Guo, L. Li, and B. He (2012). Application of 3D GIS to 3D Cadastre in Urban Environment. 3rd International Workshop on 3D Cadastres: Developments and Practices, Shenzhen, China.

BIOGPRAPHICAL NOTES

Alexandra Ribeiro holds an MSc in Civil Engineering from the University of Coimbra, Portugal. She works at the Dept. of Civil Engineering, Coimbra Institute of Engineering, Portugal. She is a PhD student in Information Science and Technology at the University of Coimbra, and a researcher at Centre for Informatics and Systems of the University of Coimbra. The focus of her research is Disaster and Risk Management.

José-Paulo Duarte de Almeida holds a licentiate degree in Land Surveying Engineering (UC), MSc in Civil Engineering (UC), and a PhD in Geomatic Engineering (University College London-UCL). He is a chartered Geomatic Engineer and has been working at UC since 1994 where he is currently lecturer in Geomatic Engineering. Following his PhD (which specialised in GIS algorithm design for the interpretation of unstructured geospatial data), he has been working on 3D cadastral modelling and systems, and is also interested in the integration of volunteered geographic information into spatial data infrastructures, such as 3D cadastral systems.

Claire D. Ellul spent 10 years as a GIS consultant in the UK, Europe and the Middle East before returning to academia in 2003. On completing her PhD in 3D GIS in 2007, she spent 2 years as a post-doctoral researcher and is now a Lecturer in Geographical Information Science at University College London. She is founder and Chair of the UK Association of Geographic Information's 3D Specialist Interest Group.

CONTACTS

Alexandra Ribeiro
Dept. of Civil Engineering
Coimbra Institute of Engineering
Rua Pedro Nunes, Quinta da Nora
3030-199 Coimbra
PORTUGAL
Tel.: +351 239 790 200
Fax: +351 239 790 221
E-mail: alexr@isec.pt

José-Paulo Duarte de Almeida
Geomatic Engineering Lab.
Dept. of Mathematics
Faculty of Science & Technology
University of Coimbra
Apartado 3008
3001-501 Coimbra
PORTUGAL
Tel.: +351 239 701 150
Fax: +351 239 793 069
E-mail: uc25666@uc.pt
Website: http://apps.uc.pt/mypage/faculty/uc25666/en

Claire D. Ellul
Department of Civil, Environmental & Geomatic Engineering
University College London
Gower Street
London WC1E 6BT
UK
Tel.: +44 (0)20 7679 4118
E-mail: c.ellul@ucl.ac.uk
Website: http://www.ucl.ac.uk/spacetimelab/people/claire-ellul