Generation of models : from urban simulation to virtual reality.

Jacques Zoller, Jean-Louis Maltret, Kai Poutrain
[jz|jlm|kap]@gamsau.archi.fr

gamsau - URA CNRS 1247
School of Architecture
184 Avenue de Luminy
F-13288 Marseille Cedex 09
Tel : +33 4 91 82 71 64
Fax : +33 4 91 82 71 71

1. Introduction

The aim of researches conducted within **gamsau** about urban simulation, in particular Remus project, is to allow rapid modeling of large and regular urban zones, for purpose of interactive navigation (like VRML) or for realistic rendering (ray-tracing methods).
One of problems to be solved in this context is the multiplicity of data formats : inputs come from different sources, and outputs are for heterogeneous systems of visualization. Typically CSG and boundary representation must be generated, treated and converted during building of models. Furthermore, the generated models can be more or less refined, depending on requests and type of use.
This paper describes the general context of data models conversion, problems concerning levels of detail and implementation done in Remus, based on object oriented approach.

2. Context of REMUS project

Urban simulation, either for historical or forecast purposes, needs many different sorts of information to produce good quality realistic images. Often, this information come from heterogeneous software environments and are of different types. Some are not directly known but must be estimated or generated for simulation, in particular information concerning buildings and urban furniture.
The problems we have to solve are the conflicting objectives of the system :

- best rendering,
- easiness of virtual walking,
- global realistic view.
- faithfulness of architectonic elements.

Some city modeling systems use existing maps and urban data as complete as possible : this allows full reconstitution compatible with reasonable size and time. Remus system, on the contrary, is able to treat approximate plans and imprecise typology in automated reconstitution.

3. Conversion between representation schemes

The need of being able to convert between different representation schemes has been often quoted in the literature [Req 80,RV 83,Mil 89]. Among all the different existing schemes, the most widely used are certainly Constructive Solid Geometry (CSG) which describes solid objects as regularised boolean combinations of solid primitives (sphere, cylinder, box, or more generally half-space), and Boundary representation (B-rep) in which objects are represented by a set of elements composing their boundary.
The conversion from a CSG representation to other schemes have been well studied. Almost all algorithms are based on the *set membership classification* [Til 80, VR 80]. Much work has been done concerning the conversion from CSG to B-rep, also known as *boundary evaluation* [Voe 78,Til 80,VR 80].

Although this is a non trivial problem, implying set operations on B-rep models known to be computationally expensive and subject to numerical problems [Man 88] many practical methods have been proposed.

The conversion from B-rep to CSG, also called *inverse boundary evaluation*, has not been paid as much attention. Moreover, converting an object represented by its boundaries to a CSG tree involves problems which are much more complex to solve. One of the principal difficulties is due to the non-uniqueness of the CSG representations.A single object can be represented by several CSG trees. Peterson [Pet 84] considered the case of simple solid polyhedra such as pyramids and extrusions. He showed that this kind of object admits a CSG representation with a monotone boolean formula (not complemented), in which each half-space, supporting the faces of such a polyhedron appears only once. He defined the so called *Peterson-style* formula which has been used in several following works.

Unfortunately, as shown by Dobkin et al. [DGH 88], all polyhedra do not admit such a Peterson-style formula. Juan-Arinyo [Ari 89] proposed a *convex-hull decomposition* for isothetic polyhedra based on Woo's *Alternating Sum of Volumes* [Woo 82], and solved the non-convergence problem encountered with *cylcic polyhedra* [Ari 95].

Whereas most of the above methods are treating two-manifold polyhedral objects, Shapiro et al. [SVV 91] applied their *boundary-based separation* to solids bounded by general quadric surfaces.

The case of general B-rep to CSG conversion is still an active research area, especially concerning the theoretical aspects. A number of practical solutions have been proposed in the two-dimensional case, and some of the results allowed the generalisation of this problem to higher dimension.

However, the domain of objects which can be handled by those algorithms needs to be extended as the type of objects used in fields like architecture are much more complex than simple polyhedra.

4. Techniques to manage levels of details

The objective of interactive navigation in a complex urban scene or inside a complex monument leads to several problems :

- impossibility of loading the complete scene, commonly several tenths or hundreds of thousands primitives,
- difficulty of treating simultaneously the most precise geometric details and the correct textures corresponding to existing materials.

These problems are usually solved by approximations and compromises :

- hierarchical structuration of the scene, allowing interactivity with a limited part of the model,
- replacement of small details by adapted textures, with loss of architectural modeling,
- adaptative refinement of polyedra, furnishing a reduced number of polygons,
- usage of poor materials, improved by tricks at the browser level.

Furthermore, in all cases, realistic lighting is not possible and is replaced by an approximate rudimentary system of lights.

For our models we have chosen to implement some techniques aiming at constructing correct architectural models giving the best realistic impression for users at any distance and scale. One of them, "reduction to primitives", is related to the fact that basic primitives of a description language are generally better treated by any browser than complex arrangements of facets. So a transformation is done from a full model to a reduced one, consisting mainly of easily visualizable simple primitives.

For the case of automatically generated models the output can be chosen at different levels of detail, either by choosing the fixed or adaptative level of meshing, or by generating the

complete correct primitives only when needed and approximate ones for saving size and complexity.
For B-rep classical models, on the contrary, the structuration is already done by the designer : the elements of a scene, named LOA (Logical Architectural Object), correspond to significative parts of a building, and can be reduced to a simpler form when it is not necessary to visualize them in full detail.

For example, among researches within **gamsau**, Roman Circus of Arles, a complex monument, has been completely modeled in a first stage with Explore software to produce high quality images. In a second step the geometric model has been converted to VRML1.0 in order to be used in a larger manner. In this case the method of reduction is based on computations of moments of inertia using Huyghens' theorem and algebraic additivity of individual moments for each part of an LOA. The original model being polyedral it is easy to sum moments of inertia and to obtain principal axes of inertia. Then the best adapted primitive is chosen, together with its characteristic parameters, by an optimization method using the geometric data. This technique of "best approximant primitive" has several advantages compared to ordinary bounding boxes : best realism by choosing an appropriate geometric form, automatic dropping of small and thin details outside main body of elements, conservation of principal geometric dimensions, physical adjustment of significative directions, easy extension to new or more complex primitives.

5. Implementation in REMUS

The representation of urban and architectural objects uses different type of modeling according to the level of detail asked. With large scale, simple polyhedra suffice to describe simplified forms or bound of architectural constituents. With small scale, objects are formed of a great quantity of elements with complex geometry, to reproduce the details of the real forms. Nevertheless there are the same relationships in assembling the different parts of the model at each level. It is therefore possible to describe them in an urban or architectural object with the use of a simplified modeler. These relations of assembling, that are of architectural nature, can be translated into simple geometrical actions : (*put_on*, *place_underneath*, *stick_on*, etc). If the information about the real object represented in this simplified model is kept by the system, a more complex representation can be produced later.

The choice in Remus is to develop a modeler based on polyhedra with B-Rep model, wich allows to obtain a simplified representation of urban or architectural elements, in relation with a description of the architectural elements themselves, then export in different formats according to needs :

- CSG representation and ray tracing renderer to produce highly realistic images
- VRML representation extracted from the B-Rep model to have real time interactive navigation.

To obtain this result, one of the main specificity in our approach is the object oriented context.

5.1 The Smalltalk environment

In Remus we use Visual Works 4.1 [BS 96], an advanced environment for Smalltalk 80 [GR83]. One of the main characteristics of Smalltalk is that everything is object. “ *Objects are collections of operations that share a state* ” [Weg 90]. The description of an object is done by classes that describe data records and algorithms relative to these data. Every object is instance of a class. An instance has characteristic variables whose values are specific to objects. The instance methods describe the algorithms which can be applied to the instances. They are used to modify or examine instances.

5.2 Class hierarchy

A tree describes the relationship class-subclass. This hierarchy on the classes is used in the mechanism of inheritance. It allows the reuse of the behavior of a class in his subclasses. This tree, corresponding to the concept "Kind Of" of general object oriented approach, allows to manage inheritance about all variables or methods. Sending messages allows object interaction.

An other main concept is the abstract classes that allow to describe general properties shared by a set of objects. There is no instance of the abstract classes but only of their sub-classes.

The development of an application in Smalltalk consists in :

- using existent classes to create instances and send them messages,
- extension with specific classes,
- modification of existing classes by adding methods.

With this approach we have implemented classes :

- to describe architectural objects,
- to describe geometry,
- to manage assembly of objects,
- to use matrix calculus for transformations,
- to display view of objects,
- to manage user interaction with all these views,
- to save and read volumes,
- to represent and treat constraints between two objects.

5.3 Description of the architectural objects

Research on architectural vocabulary has allowed to describe a model according to :

- the type of object,
- the relations between the objects,
- the attributes of the objects.

An architectural object is an instance defined by several attributes. Some are defined in the root class " *elementCorpus* ", and transmitted to all subclasses by the inheritance. The number of attributes increases with the specialization of the objects. At the root level we find :

- dimensions (height, width, length),
- typology,
- 3D morphology (geometric primitive instance of the modeler or assembly of 3D volumes),
- owner which allow in relation of composition and transversal inheritance,
- list of elements (inverse of owner with relation of composition).

The architectural typology appears in remus as a sequence of " key-value " which are associated with base of rules in the expert system that adds to the definition given for typology.

5.4 Geometric data structures

The abstract class GrObject describes the common properties for all objects that can be displayed in a view, among these, the following geometric classes have been implemented : Point3D, Axes, Plane, Face, Vol3D, Polyhedron etc.

Point3D	describes the points and vectors in three-dimensional space with homogeneous coordinates The implemented methods allow mathematical operations on points and vectors, and all others like input, output etc.
Face	Describe the polygonal outline of a polyhedral face. This description contains a list of pointers to the vertices. Its subclasses (*FreeFace*, *HollowedFace* ..) allow to transgress the Requicha rules for the good three-dimensional volumes an permit to obtain, with messages like extrude, volumes with holes
Vol3D	It is an abstract class which allows to describe 3D objects behavior with many possibilities. This allows to have a homogeneous common behavior for all 3D objects. Sub-hierarchies refine this by specific treatments, such as transformations management in terms of constraints, with the guarantee that all objects have a minimum behavior.

Polyhedron Describes the polyhedral objects behavior. The instance variables *nPoints*, *nFaces*, *pointsList*, *FacesList*, ... contain the polyhedron characteristics. The methods describe general behavior of polyhedral objects. Particular polyhedrons and their specific behaviors are described in subclasses (Cube, Box, Parallelepiped, Prism, Pyramid, ...).

5.5 Assembly data structures

Among the many possible data structures in Smalltalk, we chose to use the Smalltalk Dictionary to group our 3D objects, i.e. with couples <keys values>. A dictionary may contain either 3D objects or other dictionaries. A transformation stack is associated to the dictionary and may be applied to all its objects. Each dictionary can be seen as an union of the objects included. The property of drilling can be given to an object for the difference, but as the modeler is rough, this operation is available with two primitives only.

5.6 Export

When an architectural object receives a message to give its morphology it answers :

- the simplified internal B-rep model, for the local view,
- a text for the pov parser for realistic view ; if there is a description associated with the architectural object (text or library element with parametric description) it is exported, otherwise the three-dimensional morphology receives the message to produce the pov description,
- a production of vrml files for real time interactivity ; this is now delegated to the internal B-rep representation, and the levels of details will be treated by a specific software.

References

[Ari 89] Juan-Arinyo, R. - *On boundary to CSG and extended octrees to CSG conversions*, - pp 349-367. Springer-Verlag, 1989.

[Ari 95] Juan-Arinyo, R. - *Domain extension of isothetic polyhedra with minimal CSG representation*. - Computer Graphics Forum, december 1995, vol.14, no 5, pp 281-293.

[BS 96] X Briffaut - G Sabah, *Smalltalk - Programmation orientee objet et developpement d'applications* - Eyrolles 1996

[DGH 88] Dobkin, D., Guibas, L., Hershberger, J. and Snoeyink, J - *An efficient algorithm for finding the CSG representation of a simple polygon*.- SIGGRAPH '88, Atlanta, Georgia, august 1988, pp 31-40.

[GR 83] A Goldberg - D Robson, *Smalltalk-80 The language and its implementation* - Addison Wesley Publishing Company 1983

[Man 88] Mantyla, M. - *An Introduction to Solid Modeling*. - Computer Science Press, Rockville, Md, 1988.

[Mil 89] Miller, James R. - *Architectural issues in solid modelers.* - IEEE Computer Graphics and Applications, septembre 1989, vol.9, no 5, pp 72-87.

[Pet 84] Peterson, D.P. - *Halfspace representations of extrusions, solids of revolution and pyramids*. - tech. report. SAND84-0572, Sandia National Labs, Albuquerque, New Mexico, 1984.

[Req 80] Requicha, A. A.G. - *Representations for rigid solids: Theory methods, and systems*. - ACM Computing Surveys, december 1980, vol.12, pp 437-464.

[RV 83] Requicha, A - Voelcker, H.B. - *Solid modeling: Current status and research directions*. - IEEE Computer Graphics and Applications, october 1983, vol.3, pp 25-37.

[SVV 91] Shapiro, Vadim and Vossler, Donald L. - *Boundary-Based Separation for B-rep -> CSG Conversion*. - Tech. Report CPA91-5, Cornell Programmable Automation, july 1991.

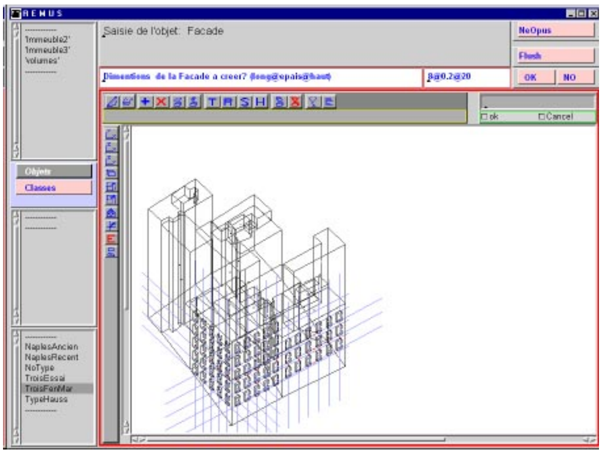
[Til 80] Tilove, R.B. - *Set membership classification: a unified approach to geometricintersection problems*. - IEEE Trans. Comput., october 1980, vol.C-29, pp 874-883.

[Voe 78] Voelcker, H., et al. - *The PADL-1.0/2 system for defining and displaying solid objects*. - In : Computer Graphics (SIGGRAPH '78 Proceedings). pp 257-263.

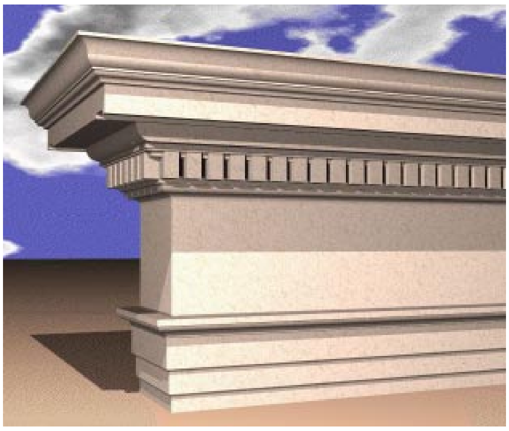
[VR 80] Voelcker, H.B - Requicha, A. - *Boundary evaluation procedures for objects defined via constructive solid geometry*. - Tech. memo 26, Production automation project, Univ. Rochester, NY 1980.

[Weg 90] Peter Wegner *Concepts and paradigms of Object Oriented Programming* - OOPS Messenger vol1 Num 1 August 1990 - ACM Press

[Woo 82] Woo, T. - *Feature extraction by volume decomposition*. - CAD/CAM Technology in Mechanical Engineering, March 24-26 1982, pp 76-94



Remus Environment


Pov Export (*Modelisation Jung Ching*)


Pov Export *(Modelisation Jung Ching*)