

Declarative Modelling in Computer Graphics: Current Results and Future Issues

Pierre-François Bonnefoi, Dimitri Plemenos, and William Ruchaud

In alphabetic order
University of Limoges, MSI laboratory,
83, rue d'Isle,
87000 Limoges, France
{bonnefoi, plemenos, ruchaud}@unilim.fr
<http://msi.unilim.fr/>

Abstract. A review of declarative scene modelling techniques is presented in this paper. After a definition of the purpose of declarative modelling, some existing declarative modellers are classified according to the manner to manage imprecision in scene description. The aim of this paper is to show the importance of declarative scene modelling for a really computer aided design and some open research problems in order to improve drawbacks of this modelling technique. Some suggestions for possible future extensions of declarative modelling are also given.

1 Introduction

Scene modelling is a very difficult task in computer graphics as traditional geometric modellers are not well adapted to computer aided design. With most of the current modelling tools the user must have quite precise idea of the scene to design before using a modeller to achieve the modelling task. In this manner, the design is not a computer aided one because the main creative ideas have been elaborated without any help of the modeller.

The problem with most of the current scene modellers is that they need, very soon during the modelling process, low-level details which are not important in the creative phase of design. This is due to the lack of levels of abstraction allowing the user to validate general ideas before resolve low-level problems. If the initial very general idea of the designer is, for example, to design a scene comporting a house, a swimming tool in front of the house and a tree on one side, this idea may be realised in many different manners. As the modeller does not offer the user an easy manner to try and test different manners to realise the initial mental idea, he (she) generally tries a small number of possible solutions and chooses the best one. In this manner, the user may lack very interesting possible solutions.

Declarative modelling tries to give intuitive solutions to this kind of problem by using Artificial Intelligence techniques which allow the user to describe high level properties of a scene and the modeller to give all the solutions corresponding to imprecise properties.

2 What Is Declarating Modelling

Declarative modelling [1, 2, 3, 4, 17] in computer graphics is a very powerful technique allowing to describe the scene to be designed in an intuitive manner, by only giving some expected properties of the scene and letting the modeller find solutions, if any, verifying these properties.

As the user may describe a scene in an intuitive manner, using common expressions, the described properties are often imprecise. For example, the user can tell the modeller that “the scene A must be put on the left of scene B”. There exist several possibilities to put a scene on the left of another one. Another kind of imprecision is due to the fact that the designer does not know the exact property his (her) scene has to satisfy and expects some proposals from the modeller. So, the user can indicate that “the house A must be near the house B” without giving any other precision. Due to this lack of precision, declarative modelling is generally a time consuming scene modelling technique.

There exist two kinds of geometric modellers, general purpose modellers, allowing to design almost everything, and specialised (or dedicated) modellers, offering high level modelling for limited specific modelling areas. In the same manner, there exist two families of declarative modellers: *general purpose* modellers, covering a large set of possible applications, and *dedicated* modellers, covering a specific area (architectural design, mechanical design, ...).

The principle of dedicated modelling is to define a declarative modeller each time it is necessary for a well delimited modelling area. Thus, PolyFormes [5] is a declarative modeller designed to generate regular or semi-regular polyhedra. The main advantage of the dedicated declarative modellers is efficiency because their solution generation engine can be well adapted to the properties of the specific modelling area covered by the modeller. On the other hand, it is difficult for such a modeller to evolve in order to be able to process another specific modelling area.

The aim of the general purpose modellers is generality. These modellers include a solution generation engine which can process several kinds of properties, together with a reduced set of pre-defined properties, as general as possible. General purpose declarative modellers could normally be specialised in a specific modelling area by adding to them new properties, corresponding to the specific modelling area we want to cover. In this sense, general purpose modellers can be seen as platforms to generate dedicated declarative modellers. The main advantage of general purpose declarative modellers is generality which allows to specialise a modeller in a specific modelling area without having to modify its solution generation engine. On the other hand, general purpose modellers suffer from their lack of efficiency, because of the generality of the solution generation mechanism. The declarative modeller MultiFormes [2, 3, 9, 10] is a general purpose declarative modeller.

It is generally admitted that the declarative modelling process is made of three phases: the *description* phase, where the designer describes the scene, the *scene generation* phase, where the modeller generates one or more scenes verifying the description, and the *scene understanding* phase, where the designer, or the modeller, tries to understand a generated scene in order to decide whether the proposed solution is a satisfactory one, or not.

3 Imprecision Management in Declarative Modellers

Declarative modelling tries to help the scene designer by allowing intuitive descriptions using a “language” close to the user’s one. This kind of description is very often imprecise and can produce many solutions. The modeller has to manage this imprecision in the scene generation phase. Two modes are used by declarative modellers to manage imprecision during the generation phase: *exploration* mode and *solution search* mode.

In exploration mode, the declarative modeller, starting from a user description, performs a full exploration of the solution space and gives the user all found solutions. This mode can be used when the designer has insufficient knowledge of a domain and wants to discover it by an exhaustive exploration or when the designer is looking for new ideas and hopes that the modeller could help him (her) by exploring a vague description. The use of imprecise properties increases the richness of the solution space and allows the user to obtain concrete answers for a vague mental image. So, the use of imprecise properties is very important for the designer. As the exploration mode is based on the use of imprecise properties, it is very important to have techniques to reduce exploration cost by reducing the number of useless tries during the solution search process [9, 10]. A problem with the exploration mode is that the use of general imprecise properties can produce a very important number of solutions and make very difficult the management of these solutions. Furthermore, some families of solutions can be of no interest for the designer and he (she) would like to avoid generation of such solutions in subsequent generations. As the modeller does not know the designer’s preferences, interaction is necessary to learn it what kind of scenes are not interesting.

In solution search mode, the modeller generally generates only one solution. To do this, either it interprets the designer’s description in a restrictive manner or it asks the designer to precise his (her) choice. So, the designer must have a relatively precise idea of the kind of scenes he (she) would like to obtain.

Declarative modellers working in exploration mode are, generally, able to work in solution search mode if the designer would like to obtain a solution immediately or very quickly from a description using less imprecise properties. As the semantic of a property is often ambiguous and several solutions not satisfactory for the user can be faced by the modeller, the designer must have the possibility to interactively intervene in order to guide the modeller in its search. So, if parts of a solution proposed by the modeller are close to the idea the designer has of the wished scene, the designer should be able to tell the modeller not to modify these parts in proposed solutions. This interaction decreases the solution space because, for a great number of scenes verifying the properties of the initial description, these parts will not satisfy the intuitive idea of the user and these scenes will be avoided.

4 A Classification of Declarative Modellers

In this section will be presented some experimental declarative modellers developed since 1987, date of the beginning of declarative modelling. These modellers will be classified according to the mode - exploration mode or solution search mode - of imprecision management they can support during the scene generation phase.

4.1 Modellers Using Exploration Mode in Scene Generation

PolyFormes [5]. The goal of the PolyFormes declarative modeller is to generate all regular and semi-regular polyhedra, or a part of the whole, according to the user's request.

Requests may be more or less precise and are expressed using dialog boxes. This initial description is then translated in an internal model which will be used during the generation process. This internal model is a knowledge base, made of a rule base and a fact base.

The scene generation process, uses an inference engine which applies rules of the rule base to the facts of the fact base and creates new facts. A solution is obtained by saturation of the fact base. The whole modeller is an expert system on polyhedra. When the initial description is imprecise, all the possible solutions are generated.

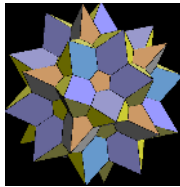


Fig. 1. Scene generated by PolyFormes

In Fig. 1 one can see an example of polyhedron generated by the PolyFormes declarative modeller. PolyFormes is a dedicated declarative modeller as it is specialised in generation of polyhedra.

MultiFormes. The first version of MultiFormes [2] was available in 1991. Several other versions of the modeller have been developed later. The purpose of this modeller was to be a general purpose declarative modeller, able to be specialised in any particular area.

MultiFormes is based on a new conception and modelling technique, declarative modelling by hierarchical decomposition (DMHD). The DMHD technique can be resumed as follows:

- If the current scene can be described using a small number of predefined high level properties, describe it.
- Otherwise, describe what is possible and then decompose the scene in a number of sub-scenes. Apply the DMHD technique to each sub-scene.

Descriptions in MultiFormes are expressed by means of dialog boxes allowing to represent a tree-like structure, to select nodes and to assign them properties. The initial description is then translated to an internal model to be used during the scene

generation process. In the first version of MultiFormes the internal model was a knowledge base made of a set of rules and a set of facts. In all the other versions of the modeller, the internal model is a set of arithmetic constraints on finite domains (CSP). The scene generation process uses a constraint satisfaction engine which applies CSP techniques [14] to generate all the solutions corresponding to a given description. A special form of primitive arithmetic constraints, CLP (FD) [15], is used to improve the scene generation process.

The tree of the hierarchical description of a scene, used in the scene generation phase, allows scene generation in various levels of detail and reduction of the generation's cost. To do this, the modeller uses a bounding box for each node of the tree. This bounding box is the bounding box of the sub-scene represented by the sub-tree whose the current node is the root. All bounding boxes of the children nodes of a node are physically included in the bounding box of the parent node. This property permits to detect very soon branches of the generation tree which cannot be solutions. In Fig. 2, the spatial relation between the bounding boxes of a scene and its sub-scenes is shown (left), as well as a scene generated by MultiFormes (middle).

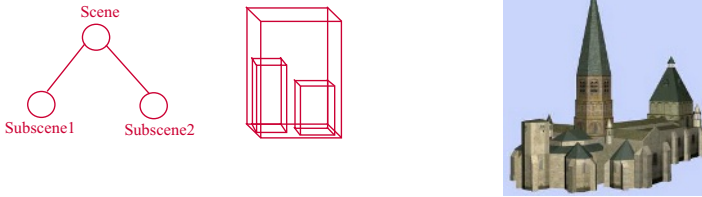


Fig. 2. From left to right: the bounding boxes of the sub-scenes of a scene are inside the bounding box of the parent scene; inside a 3-floor building; Cathedral of Le Dorat (France) designed by W. Ruchaud

MultiFormes is a general purpose declarative modeller which can be specialised by adding new predefined properties. The scene generation phase works in exploration mode, whereas it is possible to use solution search mode by means of user's interaction. MultiFormes can also use a set of geometric constraints [9] instead of arithmetic ones. This set contains constraints like "point P is in the box B" or "Points P1, P2, P3 are aligned". Satisfaction of this kind of constraints is computed using CSP-like techniques and allows more flexibility in creating scenes. An example of scene generated by geometric constraint satisfaction techniques can be seen in Fig. 2. Machine learning techniques based on neural networks [6, 7] have been implemented in MultiFormes [8]. These techniques allow the modeller to select scenes close to the designer's desires in solution search mode, during the scene generation phase.

4.2 Modellers Using Solution Search Mode in Scene Generation

DE2MONS. The DE2MONS declarative modeller [19] is a general purpose modeller whose main properties are:

- A multi modal interface,
- A generation engine limited to the placement of objects,
- A constraint solver able to process dynamic and hierarchical constraints.

The modeller uses a multi modal interface allowing descriptions by means of the voice, the keyboard (natural language), a data glove or 3D captors informing the system of the user's position. The description is translated in an internal model made of linear constraints. The generation engine of DE2MONS uses a linear constraint solver, ORANOS, able to process dynamic constraints (new constraints can be added during generation) and hierarchical constraints. Hierarchical constraints are constraints with priorities assigned by the user. Whenever there is no solution for a given description, constraints with low priority are released in order to always get a solution. The solver computes one solution for a given description.

CCAD. The Cooperative Computer Aided Design (CCAD) paradigm was introduced by S. Kochhar [11,12] to facilitate the integration of *generative* and traditional modelling systems by allowing the designer to guide the generative system through successive rounds of automated geometric modelling. The notion of generative modelling is very close to the notion of declarative modelling, as in both cases imprecise descriptions can generate many solutions. An experimental cooperative scene modeller was implemented for a generative system based on the formal language of schema grammars. The CCAD framework is based on three main premises:

- A generative geometric modelling (GGM) system exists and can be used to generate a set of designs based on some designer-defined constraints or properties.
 - The GGM system is supposed not to produce perfect designs, but rather it will be guided to search for better designs by the human designer.
 - As the GGM system produces a large set of designs, a specialised browsing system allows the designer to search the set of generated designs in a directed manner.
- A typical modelling session using the CCAD system proceeds as follows:
- The designer uses the TGM system to generate a nascent design to be used in the first iteration of automated modelling.
 - The designer then uses a dialog with the GGM system to define the constraints to be used during the generation process.
 - The GGM system then instantiates all valid geometric designs. These designs are presented as icon-like buttons in a large screen area and the designer can get a large image of a design by clicking on the corresponding button.
 - The designer then selects a set of promising designs using the browsing system.
 - The selected designs are then returned to GGM system and the 4 four last steps are repeated until the desired design has been constructed.

The CCAD paradigm has also been applied by D. Sellinger [13] in a PhD thesis whose purpose was to integrate declarative and traditional geometric modelling.

VoluFormes [16]. VoluFormes is a *dedicated* declarative modeller allowing the user to quickly define boxes in the space whose purpose is to check the growth of forms. It is made of two modules:

- *Voluboites*, which allows to define boxes where the spatial control is performed.
- *Voluscenes*, which allows to use growth mechanisms applied to elementary germs and to create forms, taking into account the spatial control boxes.

Only Voluboites works in declarative manner. The positions of spatial control boxes are described during the description phase using a natural-like language. Description and generation work in incremental manner. Each box is placed in the 3D space and, if the user does not like the proposed box and placement, another solution

can be given. Once the current box is placed in the space, the same process is applied to define the next one. The generation engine of Voluboites uses a CSP-like constraint satisfaction technique. On the left of Fig. 10, one can see a boxes arrangement obtained by Voluboites.

Voluscenes is used by the designer to put germs in the boxes and to select the growth mechanism among a reduced number of predefined imperative mechanisms. On the right of Fig. 3 one can see an example of scene obtained by form growth.

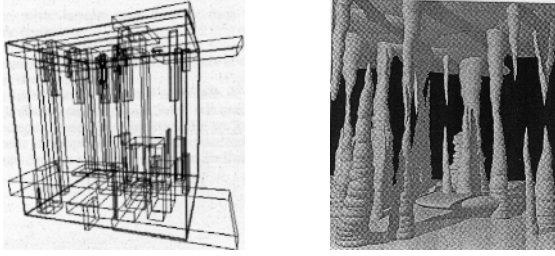


Fig. 3. Boxes arrangement and form growth with VoluFormes

5 Scene Understanding in Declarative Modelling

As declarative modelling generates several solutions and most of them can be unexpected, it is often necessary that the modeller offers scene understanding techniques in order to allow the designer to verify the properties of an obtained solution. Scene understanding can be visual or textual. Most of existing declarative modellers use simple scene display from an arbitrary chosen point of view. Very few declarative modellers use sophisticated scene understanding mechanisms.

PolyFormes uses a “matches-like” display mode allowing the user to better understand the shape of a generated polyhedron. In this kind of display, only the edges of the polyhedron are displayed but they are thickened (see Fig. 4).

MultiFormes uses more sophisticated techniques for scene understanding [20]. These techniques use a good view criterion based on the scene’s geometry and automatically compute a good point of view by *heuristic search*. As a single point of view is not always sufficient to understand complex scenes, MultiFormes also proposes an intelligent automatic scene exploration by a virtual camera, moving on the surface of a sphere surrounding the scene (Fig. 4).



Fig. 4. From left to right: “Matches-like” display of polyhedra; Scene automated exploration by a virtual camera

6 Advantages and Drawbacks of Declarative Modelling

The purpose of declarative modelling is to improve the hard task of scene modelling by allowing the designer to use a high level of abstraction. Taking into account the existing declarative modellers, it is possible to extract the main advantages and drawbacks of declarative modelling.

Declarative modelling is closer to the user than traditional geometric modelling because the user has not to take into account geometric representation and construction of the scene. The declarative modeller produces a scene with the high level properties described by the designer and then translates this solution to a geometric model. In declarative modelling it is possible to describe a scene up to a chosen level of detail. In this manner, even if the designer has not yet determined some parts of the scene, he (she) can get a first draft of the scene and refine it later. Modifications of a scene are easier with declarative modelling because it is easier to replace a property by another one than to modify the scene's geometry. Indeed, it is easier to modify an object by telling the modeller that it is not enough high than by changing the list of vertices in the geometric model of the object. Declarative modelling allows really computer aided design because, from an initial description of a vague idea of the designer it is able to propose several solutions and help the designer in the creative phase of scene design. A declarative modeller can evolve because, if properties are defined in a manner not depending on the scene generation engine, new properties can easily be added to the modeller and improve its designing power.

The main drawback of declarative modelling comes from the possibility to use imprecise properties. When, for a given description, the *search space* is much wider than the *solution space*, the generation process is very time consuming. A research field in the area of declarative modelling is to find efficient methods of reducing the search space. Another drawback is due to the possibility to get many solutions from a scene description. The management of an important number of solutions is always difficult because the designer cannot remember all the solutions during the designing process. A possible solution would be to define a metric and classes of close to each other solutions by the modeller and to show only a representative scene from each class [18]. The problem is that it is not always possible to define such a metric, especially with general purpose modellers.

7 Future Issues

The problem declarative modelling has to face is a hard one. Despite of this complexity, the performances of existing declarative modeller prototypes are quite satisfactory. However, the problem of efficient reduction of the search space is an open research problem.

Current declarative modellers are essentially concerned with geometrical or topological aspects of a scene. However, it is possible to describe in declarative manner non geometric properties of the scene such as ambience (lighting, fog, ...). Some research works have started in this area. In a general manner, if a property may be

translated into constraints on numerical values, it can be described and processed in declarative manner.

Another challenge is to include time in declarative modelling, especially for scenes corresponding to existing things in the real world. For instance, it would be interesting for an architect to show a client not only the current state of a building to be designed but also its appearance 10 or 15 years later, if a small number of parameters such as, construction materials, climate, etc., are known. We call this kind of modelling *predictive declarative modelling*. Some works have started in this area too.

Finally, coupling declarative and traditional scene modelling should give interesting results because it would allow to reduce the cost of declarative modelling, by permitting to first define a draft of the scene by declarative modelling and then refine the draft using an integrated geometric modeller. Such an integrated declarative-traditional geometric modeller has already been implemented [13] but a lot of improvements are still possible in this area.

8 Conclusion

In this paper we have tried to present the challenge of declarative modelling as a tool to make easier the scene designer's work and offering a really computer aided design able to stimulate the designer's creativity. Even if many things have to be improved in declarative modelling, most dedicated declarative modellers are very efficient and, above all, able to produce scenes impossible to obtain by traditional geometric modelling. Declarative modelling has opened a lot of research fields in computer graphics as well as in artificial intelligence. Open research fields currently concern reduction of search space during the generation phase (efficient constraint resolution), management of a big number of solutions (classification, machine-learning, interactivity) and scene understanding (scene visual complexity, path optimisation).

The authors would like to thank Dr. Andres Iglesias for his kind invitation to the CGGM conference.

References

1. Lucas M., Martin D., Martin P., Plemenos D.: The ExploFormes project: some steps towards declarative modelling of forms. Published in BIGRE, no 67, pp 35 – 49, 1990.
2. Plemenos D.: A contribution to study and development of scene modeling, generation and display techniques - The MultiFormes project. Professorial Dissertation, Nantes (France), November 1991 (in French).
3. Plemenos D.: Declarative modeling by hierarchical decomposition. The actual state of the MultiFormes project. International Conference GraphiCon'95, St Petersburg, Russia, 3-7 of July 1995.
4. Plemenos D., Tamine K.: Increasing the efficiency of declarative modelling. Constraint evaluation for the hierarchical decomposition approach. International Conference WSCG'97, Plzen (Czech Republic), February 1997.
5. Martin D., Martin P.: PolyFormes: software for the declarative modelling of polyhedra. *The Visual Computer* (1999) 55-76.

6. McCulloch W.S., Pitts W.: A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5, 115 - 133, 1943.
7. F. Rosenblatt: The perceptron: a perceiving and recognizing automaton. Project Para, Cornell Aeronautical Lab. Report 85-460-1, 1957.
8. Plemenos D., Miaoulis G., Vassilas N.: Machine learning for a general purpose declarative scene modeller. *International Conference GraphiCon'2002*, Nizhny Novgorod (Russia), September 15-21, 2002.
9. Ruchaud W., Plemenos D.: MultiFormes: a declarative modeller as a 3D scene sketching tool. *Int. Conference ICCVG'2002*, Zakopane (Poland), September 25-29, 2002.
10. Bonnefoi P.-F.: Constraint satisfaction techniques for declarative modelling. Application to concurrent generation of scenes. PhD thesis, Limoges (France), June 1999.
11. Kochhar S.: Cooperative Computer-Aided Design : a paradigm for automating the design and modeling of graphical objects. PhD thesis, Harvard University, Aiken Computation Laboratory, 33 Oxford Street, Cambridge, Mass. 02138, 1990. Available as TR-18-90.
12. Kochhar S.: CCAD : A paradigm for human-computer cooperation in design. *IEEE Computer Graphics and Applications*, May 1994.
13. Sellinger D., Plemenos D.: Interactive Generative Geometric Modeling by Geometric to Declarative Representation Conversion. *WSCG'97 conference*, Plzen (Czech Republic), 10-14 February 1997.
14. van Hentenryck P.: Constraint satisfaction in logic programming. *Logic Programming Series*, MIT Press, 1989.
15. Diaz D.: A study of compiling techniques for logic languages for programming by constraints on finite domains: the clp(FD) system. PhD thesis, 1995, Orleans, France.
16. Chauvat D.: The VoluFormes Project: An example of declarative modelling with spatial control. PhD Thesis, Nantes, December 1994.
17. Poulet F., Lucas M.: Modelling megalithic sites, *Eurographics'96*, pp 279-288, Poitiers (France), 1996.
18. Champciaux L.: Introduction of learning techniques in declarative modelling, PhD thesis, Nantes (France), June 1998.
19. Kwaiter G.: Declarative scene modelling: study and implementation of constraint solvers. PhD thesis, Toulouse (France), December 1998.
20. Barral P., Dorme G., Plemenos D.: Visual understanding of a scene by automatic movement of a camera. Short paper. *Eurographics 2000*.