THE ARCHITECT AS TOOLMAKER

Computer-Based Generative Design Tools and Methods

CRISTIANO CECCATO
Visiting Fellow, School of Design, The Hong Kong Polytechnic University.
Lecturer, Architectural Association School of Architecture, London, UK.
E-mail: nautilus@bogo.co.uk

Abstract. The purpose of this paper is to illustrate the results of various stages of research into the development of generative design methods and tools, conducted at the Architectural Association School of Architecture (London), Imperial College of Science, Technology and Medicine (London), and independently. A brief introduction explains the philosophy behind generative design methods and their basic principles. A number of computer software tools and projects developed by the author are then used to illustrate the methodology, techniques and features of generative design and its organisation of information.

1. Introduction

Until recently, the architect has utilised computers as a more efficient version of the draughting table, but not as an active aide in the design process. The gesture of putting pen to paper, as it were, remains unchanged in the electronic realm: a design, pre-conceived in the mind's eye, is put to the screen by means of stylus or mouse. In these cases, the computer merely acts as a receptacle for finite ideas, or as a construction tool. The understanding of fundamental shape-forming processes in nature allows us to create tools that support our design intuition. These tools do not attempt to produce a finished solution, or re-invent draughting. Instead, they integrate into the design process at different stages. In the early phases of development they can contribute to generating a wide range of ideas, thus turning the computer into a 'muse', inspiring us with alternatives. At different stage, mathematical search and cognitive methods (such as the Genetic Algorithm, or Neural Networks) can efficiently help in the optimisation a design.

Naturally, it is up to each of us to understand in which way we would like to have a tool contribute to our design. Process-driven design is not a science, and the use of tools reflects our own individuality in design. In other words, these tools add to the architect's palette of instruments, which he can use as he pleases. The approach is extended, however, by enabling the architect to create his own tools as he requires. This is already evident today through the 'customisation' of CAD systems, but can be taken a step further through the creation of

independent, intuitive tools that complement, rather than compete with, the various existing systems [1]. At the same time, such methods can be used outside of the design field in a variety of tasks, for example, to visualise complex information.

2. Projects and Tools

The selected projects attempt to illustrate some of the ideas introduced above. Not all of the projects are conceived as functional software tools, but nonetheless contribute to showing some concepts of generative or 'intelligent' design computing at work.

The main goal of this small collection of work is to demonstrate to the reader some of the possibilities of generative design methods, and hopefully inspire an interest in pursuing similar work, or supporting the development of such tools. With time-to-market requirements in the industrial design and architectural professions becoming ever more stringent [2], and the financial need to exploit available computing power as much as possible, it seems clear that generative design tools will become central in supporting and enhancing the engineering and design professions in the future.

2.1 METABOLIC GROWTH: EMERGENT FORM THROUGH ENERGY EXCHANGES (1995)

This project is not a 'tool' as such; rather it is a pilot project devised to investigate the potential of hierarchical rule-based design systems. Past work done in Diploma Unit 11 of the Architectural Association was extremely successful in developing highly complex forms in three-dimensional datastructures [3,4]. The work was inspired by morphogenetic processes observable in nature, in particular by the phenomena of hierarchical and rule-based growth. The question is, when does data stop being 'virtual' data, and can be considered actual material form?

In nature, shells emerge almost by accident. Certain kinds of bacteria living in the sea produce calcium as a by-product of metabolism. This is crystallised and shed out of the organism's plasma so that it may avoid calcium poisoning. This simple operation has a tremendous effect: the calcium collects on the surface of the bacterium, forming a layer - a shell [5]. The transition from one- to multicellular forms did not change the situation: The skeletons of Radiolarians are examples of the complexity these forms can obtain [6]. In a sense, this by-product of the functioning and interaction of living cells could be considered plausible *dross* [7].

We attempted a project that carefully modelled the processes of energy exchanges at cellular level as well as the externalisation of excess energy as *dross*. This was achieved by building a three-dimensional *cellular automaton* engine [8]. Each cell-space interacted with its neighbours, as well as a simulated growth environment in discrete time steps [9]. The basic rules were twofold: cellular interaction with neighbours in spatial terms, and the interaction on energetic terms. In the latter, sufficient amounts of energy had to be produced by each cell in order to 'stay alive', and excess energy was externalised as 'dross', located at the cell's spatial co-ordinates. The simulated environment was also implemented as a cellular automaton in order to insure maximum unpredictability of environment behaviour.

The emergent forms were *only* controlled by the rules governing the inter-cellular activities, and by the hierarchical growth functions in the processes of ontogeny. They may thus be considered the result of collective self-organising behaviour at the level of the individual cell as well as of emergent order across multiple scales of operation. The project's value, in our view, lies in its demonstration of the power of generative rule-based design on the hierarchical level.

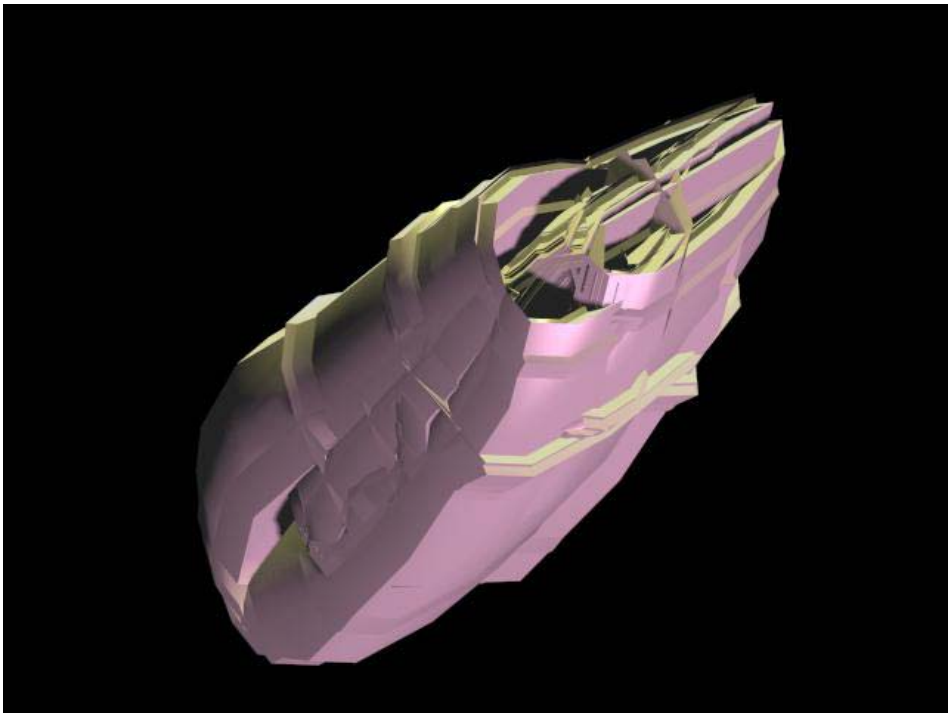

Figure 1. Emergent Form (above)

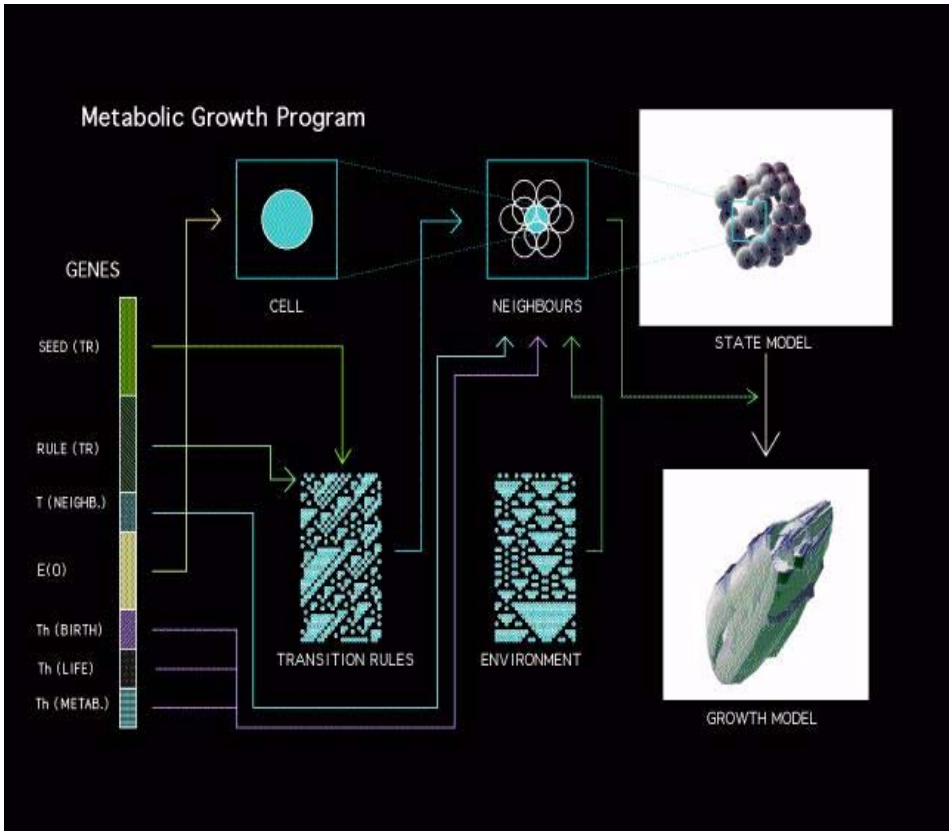

Figure 2. Growth System

2.2 MENDEL: A SCENE CONSTRUCTOR BASED ON THE GENETIC ALGORITHM (1997)

Mendel is a small, intuitive software tool, and was conceived to demonstrate the potential of generative computing as an important asset to the design process. It can be seen as a small '3D sketch tool' which allows the user to quickly generate and explore a large range of related shape configurations, simply by selecting those configurations he prefers. This sounds straightforward and almost banal, but one is struck by the sheer difficulty of operating in the way described if one pauses to consider today's most common three-dimensional software tools. These are usually extremely efficient in assisting with draughting, construction and management of large quantities of spatial and structural data, but are almost at a loss when it comes to supporting the creative design process. In other words, today's tool may form a powerful support for the *profession* of design or

architecture, but are weak in supporting its design *process* [10]. They expect a largely finished design, sketched on paper or roughly modelled in cardboard or wood, to be entered into a precise draughting and modelling system.

The question arises: what about if the designer wishes to experiment with form? What if the designer is unclear as to what his exact goal is – that is, wants to 'sketch'? This question does not call for an immediate revolution in the computer-based design process; rather, it advocates the need for an *additional* set of design tools, which *complement* and *integrate* into the existing CAD methodology. The *Mendel* project proposes such a tool: Its implementation, as illustrated in this paper, is as an independent piece of software able to exchange data with conventional CAD systems; it could equally be implemented as a plug-in device for the CAD software itself.

The project takes much inspiration from the *Biomorph* tool as devised by Prof. Richard Dawkins [11]. *Biomorph* allows the user to 'evolve' a series of 'stick-animals' based on simple rules governing size, number and ramification properties of the shapes. It uses a *Genetic Algorithm* (GA) [12,13] at its heart, as the main 'engine'. While it is not within the scope of this paper to describe in detail the nature and applications of the GA, it can be said, in simple terms, that the GA closely emulates the process of breeding and selection found in nature. While nature decrees the survival of the fittest in the natural environment, the artificial GA is in fact a powerful search engine capable of quickly finding optimal solutions in complex, non-linear data sets, thus achieving a kind of machine learning. GAs became popular in engineering [14] for the solving and optimisation of problems from pipeline systems to aircraft wings to microchip design; in the case of Mendel, the fitness criterion is not efficiency, but the designer's liking of a particular design configuration.

By picking and choosing those designs he seems 'fit' for the ideas or shapes they contain, or their appearance, the designer 'plays God' to his design population, constantly refining it according to his own vision. The pick-and-breed character of Mendel allows the designer to quickly perform what are effectively a new type of intuitive design operations on his work, without getting lost in the technical problems of accurate CAD draughting and construction.

Specifically, the user is able to build and manipulate complex 3D scenes by selecting and 'breeding' combinations of shapes to achieve new spatial layouts and structural variants. The genetic 'codescript' [15] is, for the sake of simplicity, the structure of the 3D scene itself, built as a scene-tree consisting of primitive shape-, material-, and spatial transformation-nodes; and analogous sub-trees.

The user controls the system parametrically by altering the 'genetic' breeding criteria: Mutations in the structure of the scene, as well as the individual data values defining the tree nodes. He can also perform asexual (using a single scene-structure) or sexual (using a pair of scene structures) breeding, and can always add new 'genetic material' into the process by importing new scenes into the program.

Naturally, the processes described above, as well as the earlier engineering examples show that the GA could also be employed in the field of design to solve more goal-oriented problems, such as optimising the airflow of a car's body, or circulation within a building. This furnishes the designer with a more objective set of tools, which together with intuitive design tools greatly broadens his/her computerised palette.

Mendel constructs scenes of primitive shapes. A similar program operating on free-form surfaces is in development. *Mendel* for Windows 95/98/NT4 can be obtained at:

http://www.bogo.co.uk/natuilus/mdl3209b.exe

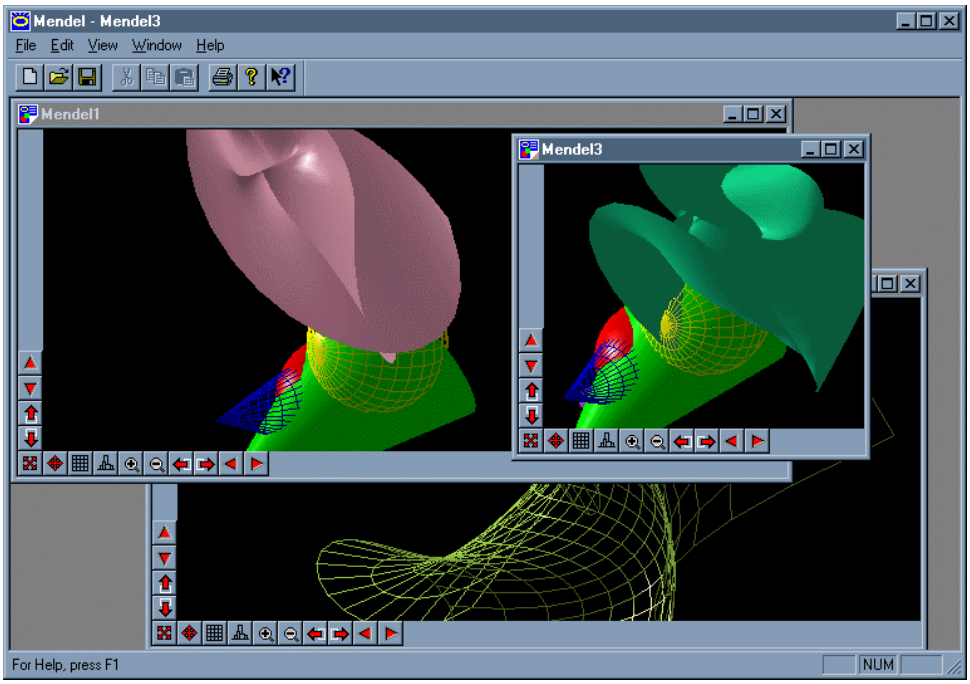

Figure 3. Mendel for complex surface forms (in development)

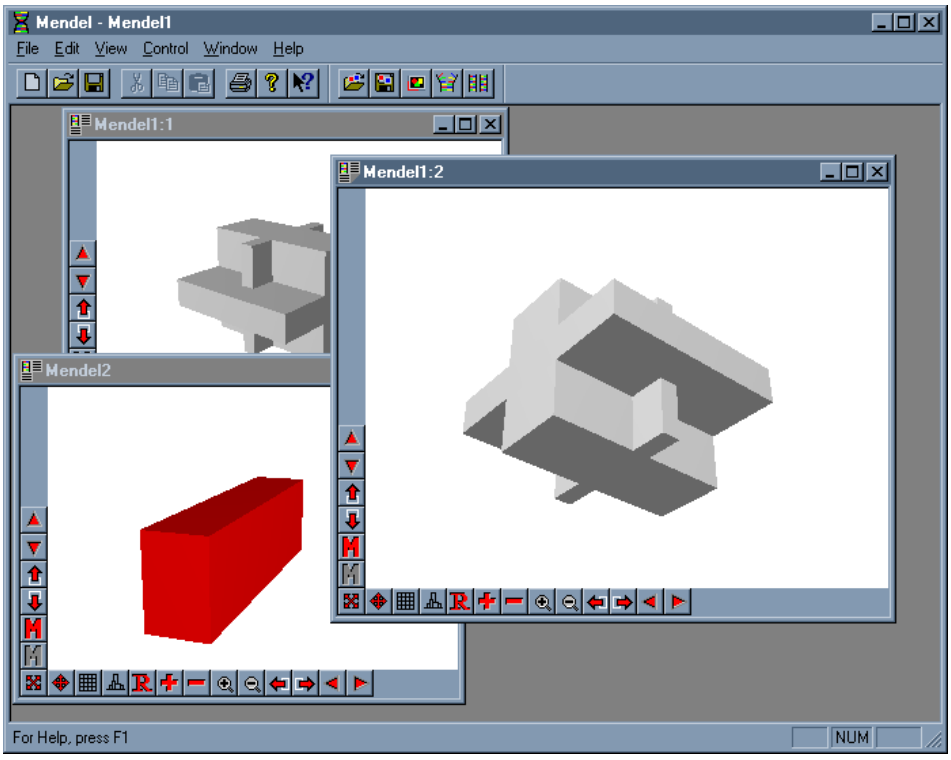

Figure 4. Mendel breeding primitives

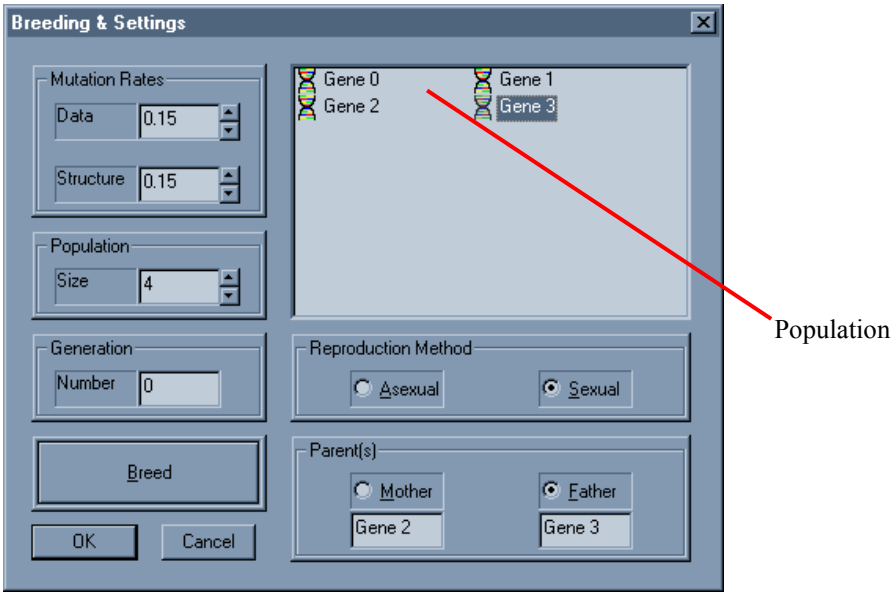

Figure 5. Breeding control dialog

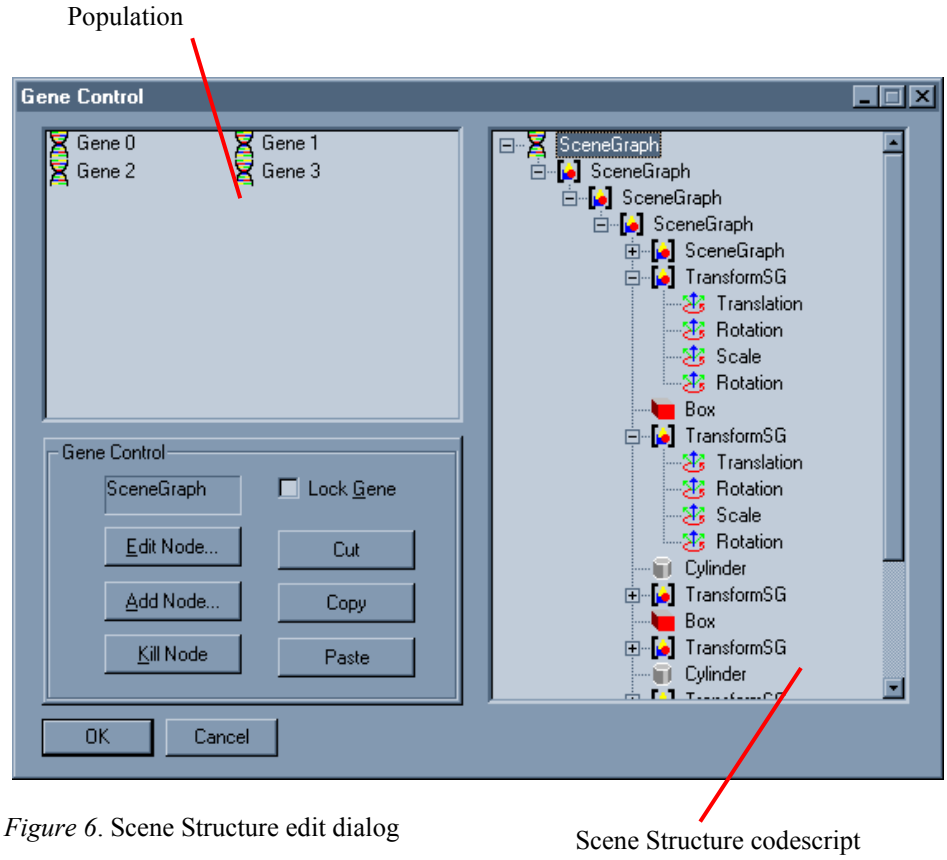

Figure 6. Scene Structure edit dialog

2.3 **MAGICBOX:** A CELLULAR AUTOMATA BASED SPACE ORGANISER (1998)

This tool was originally developed as a pilot project for Zaha Hadid Architects in London. The firm wanted to attempt employing a generative tool in a real-life situation, in this case an important architectural competition in the USA. Part of the brief called for a flexible layout of furniture and partitions in large halls, so the idea was to organise the spaces through a rule-based system. The simplest method, which we used as the engine for the tool, was a two-dimensional cellular automaton [16], which distributed furniture layouts across a floor grid according to pre-set spatial rules. Altering the rules produces different spatial layouts. In a second phase, the tool would be able to incorporate learning abilities (GA), by which to apprehend optimal layout rules. For maximum efficiency in the architectural office environment, the tool was developed as a plug-in for AutoCAD, and was able to act directly on its drawing database.

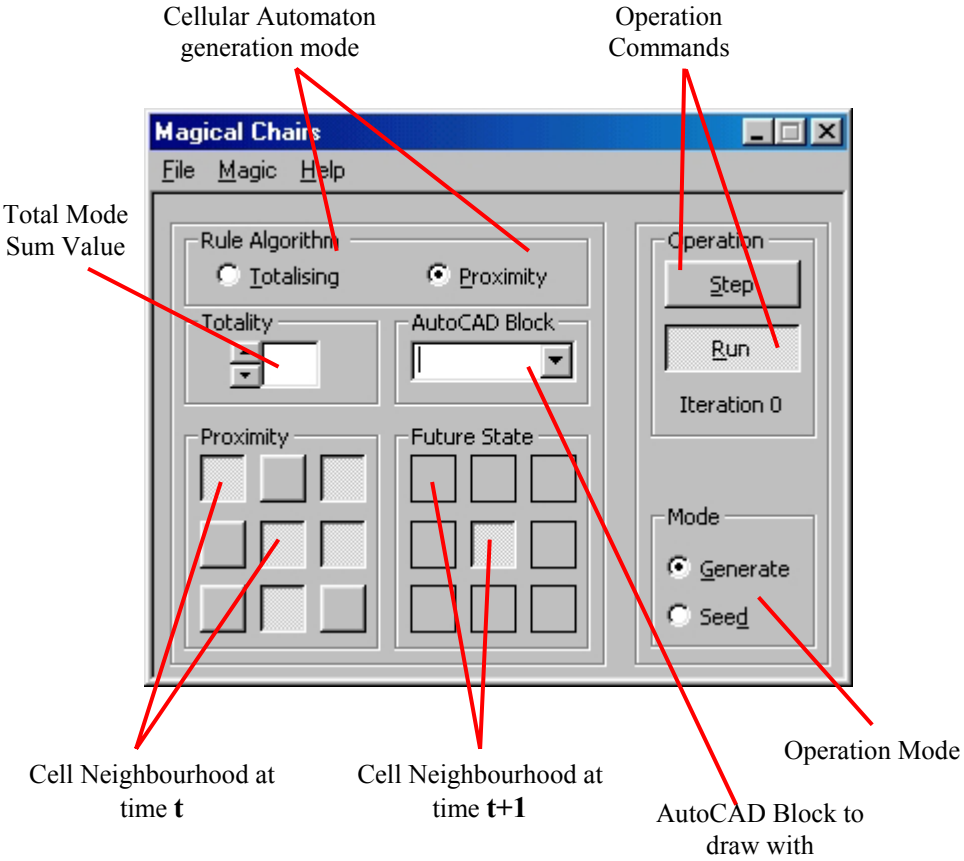

Figure 7. The "MagicBox" control dialog.

References

1. GREENBERG, S., HAYNE, S., and RADA, R., *Groupware for Real-Time Drawing: A Designer's Guide* (Maidenhead, Berks.: McGraw-Hill 1995)
2. BLYTH, A., "Managing a Global Practice." In *Architect's Journal,* No. 4, Vol. 202 (London 1995)
3. FRAZER, J. H., *An Evolutionary Architecture* (London: Architectural Association 1995)
4. FRAZER, J. H., "The Architectural Relevance of Cyberspace." In Toy, M., ed., *Archtiectural Design* (London: Architectural Design 1996)
5. SAGAN, D., "Metametazoa: Biology and Multiplicity." In J. Crary, and S. Kwinter, eds., *ZONE 6: Incorporations* (New York: Urzone 1992)
6. THOMPSON, D. W., *On Growth and Form* (Cambridge: Cambridge University Press 1961)
7. Frazer (1)

8. Ibid.
9. Ibid.
10. Greenberg
11. DAWKINS, R., *The Blind Watchmaker* (London: Penguin Books 1986)
12. GOLDBGERG, D. E., *Genetic Algorithms in Search, Optimization & Machine Learning* (New York: Addison-Wesley 1989)
13. Frazer (1)
14. Goldberg
15. Frazer(1)
16. Ibid.