STRUCTURAL SIMULATION MODELS THAT BUILD THEMSELVES

L. JANKOVIC †, S. JANKOVIC*, A. H. C. CHAN*, G. H. LITTLE*

†*InteSys Ltd, University of Birmingham Research Park, Vincent Drive, Birmingham B15 2SQ, UK.*
L.Jankovic@invirtuo.com
**School of Civil Engineering, University of Birmingham, Edgbaston, Birmingham, B15 2TT, UK.*

Abstract: The paper reports on creation of analogue models of structures in virtual reality, using an emergent, or bottom-up approach, inspired by laws of Complexity. This approach focuses on component based modelling using fundamental laws of physics. The creation of system models is then achieved through interaction of the components, from the bottom-up, leading to a self-organisation of the system model, without explicit programming. The paper describes advantages of this modelling approach in comparison with conventional top-down simulation methods, which in contrast, typically describe the entire modelled system with a system of equations, solved repeatedly in each simulation time step.

Keywords: bottom-up, emergent, analogue models, structures, complexity, self-organisation, top-down.

1. INTRODUCTION

The work reported in this paper is inspired by the need of structural engineers for better simulation and analysis tools.

Conventional analysis of structures is based on systems of algebraic equations that describing the entire modelled system, which is then solved by matrix inversion. This top-down approach is not entirely satisfactory for the changing needs of the industry. It does not allow for a time efficient dynamic review of the design in progress, and being a system rather than a component based, it does not allow for an easy real-time visualisation of the component behaviour.

This paper will present a radically different approach to modelling and simulation of structures. It will not use matrices and other mathematical tools with which structural engineers are familiar. In contrast, it will abandon conventional solution methods, and go back to the basics of how natural and man-made systems work, drawing its inspiration in the laws of Complexity [Kauffman, 1996]. The next section will first investigate the reasons for inefficiency of the conventional methods.

2. STRUCTURES DO NOT KNOW MATRIX CALCULUS

We now look more closely into the usability of conventional solution methods and their simulation capabilities.

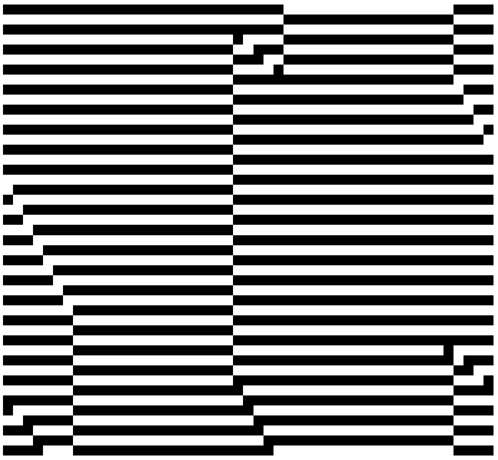
Conventional analysis of structures is based on systems of algebraic equations that need to be solved in the process of determining the design parameters for the structure. One of the most commonly used methods, the finite element method, represents each structural component with a stiffness matrix, consisting of differential equations based on Newton's laws. Assembling the matrices of individual components into a global matrix creates a model of an entire structure. The system of equations from the global matrix is then solved to obtain design parameters for the structure.

As the behaviour of the system components is only known after the system of equations has been solved, the visualisation of structural behaviour cannot be efficiently integrated into the conventional analysis tools. This problem was also confirmed independently by other authors [Connell and Tullberg, 2000], in connection with difficulties of visualisation of finite element calculations of a bridge, even when using high performance state of the art computers.

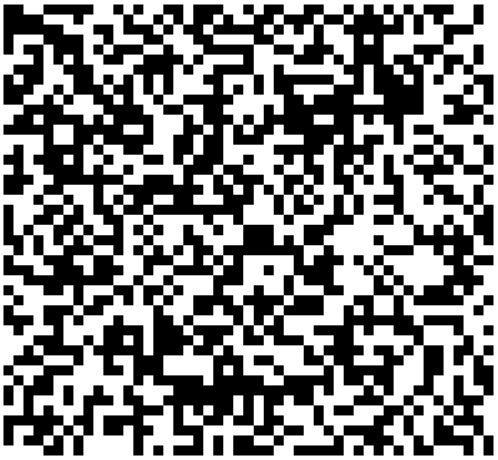
Although modern finite element packages [FEA Ltd, 1999] have comprehensive visualisation facilities based on comparison of undeformed and deformed shape of the structure, the user has to wait between the two images until the calculation process has completed. Some finite element packages offer animation as a better way to visualise the results of structural action, but even then it is not possible to change the load interactively and to observe the structural behaviour in response to the user interaction.

None of the above visualisation capabilities are entirely satisfactory, as they do not appear to support a dynamic simulation of structures. The main obstacle for the dynamic simulation and

a) Orderly phase - the cells quickly settle to a frozen state



b) Phase transition (edge of chaos) with self-organised repeating patterns travelling from left to right



c) Chaotic phase - the cells continuously change, without settling

Figure 1 A multi-component cellular system a) in orderly phase; b) in phase transition; c) in chaotic phase

visualisation capability appears to be in the solution method, based on extensive mathematical formalism that is somewhat removed from the physics of the problem. Structures stand and react to loads without knowing how to do the matrix calculus. We therefore seek a method of representation of the structure in the simulation model that is analogous to the real structure.

The next section takes us back to the basics, in an attempt to understand how structures and other systems work, which we believe would help with the creation of a dynamic simulation model.

3. BACK TO THE BASICS - BACK TO COMPLEXITY

We believe that structures, as well as a vast number of other complex systems, are driven by some very simple rules, and therefore that the analysis of such systems can be reduced to the discovery and analysis of these rules. To help us with finding the rules, we turn to the Science of Complexity - a science that investigates how simple rules applied to system components can give rise to richness and diversity of natural and man-made systems.

Complexity was first defined as the "edge of chaos" [Langton, 1992]. The edge of chaos, or complexity, is a region of "phase transition" between orderly and chaotic behaviour of a multi-component system.

We illustrate this by observing a multi-cellular system, similar to a well known game of life, in which each cell is connected to a number of other cells across the system, and in which the status of the cell (whether on or off) is determined on the basis of a simple Boolean function performed on all inputs into the cell, representing the statuses of the connected cells.

In a completely orderly, sub-critically connected system, any dynamic behaviour very quickly settles into a "frozen" state, and such system is not sensitive to initial conditions and external inputs (Fig. 1a).

The same system, but super-critically connected, operates completely in a chaotic state, and can develop unpredictable behaviour even from small changes in initial conditions and external inputs. This is a well known, and perhaps somewhat exaggerated butterfly effect, where a butterfly flapping its wings in one part of the globe, causes a weather storm in another

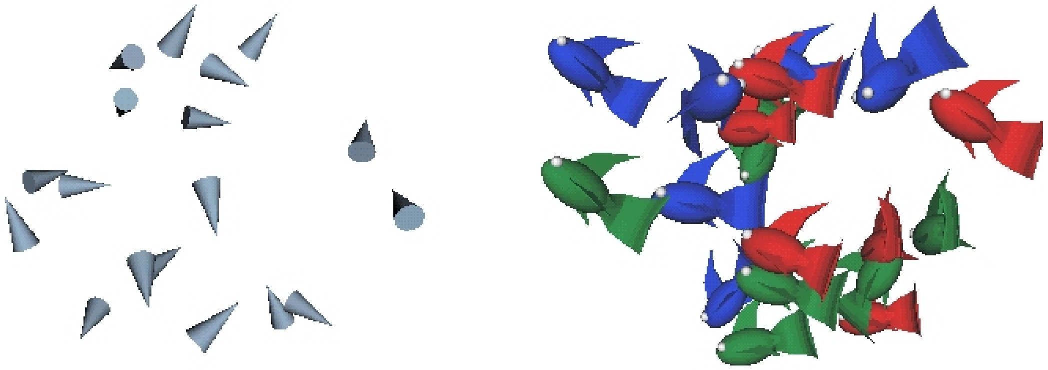


Fig. 2 An emergent model of a flock of boids (left) and school of fish (right) developed in VRML97 and JavaScript

part of the globe. A conceptual system in chaotic regime is shown in Fig. 1c.

However, if the system is critically connected (Fig. 1b), it enters the edge of chaos, or complex regime, where unpredictable but self-organised behaviour can occur. It has been shown that self-organisation and regular moving patterns in systems at the edge of chaos can support generalised computation [Langton, 1992].

[Kauffman, 1996] gives numerous examples of such systems in which the behaviour of the complex system as a whole is more than the behaviour of the sum of its parts. Through the unexpected behaviour of the system as a whole, the complex system appears to self-organise, positioning itself between order and chaos, and driven by equilibrium of energy (equilibrium systems), or stability of structures through constant supply and dissipation of mass and energy (non-equilibrium systems). By discovering laws of complexity [Kauffman, 1996] makes a serious challenge even to the well-established theory of natural selection by Darwin. In this context our re-examination of the well-established methods of structural analysis does not seem out of order.

A number of models of complex systems have been created using similar principles. Figure 2 shows an example of a dynamic model of animal movement, originally named "boids" by its inventor [Reynolds, 1987]. The rules for individual boids are that they must keep apart, keep together and keep going ahead. With only three simple rules, a very complex behaviour emerged from the bottom-up, spontaneously creating a system that cannot be modelled with a finite element method or top-down equations. The complexity of system models achieved with this approach is considerably disproportional to the simplicity of the component models. Reynolds used the same algorithm for animation of stampeding animals in Disney's Lion King.

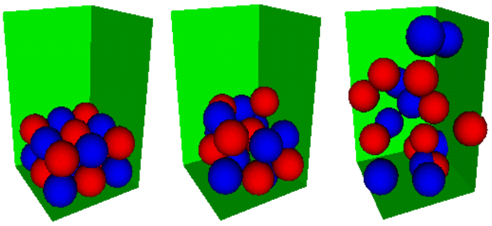


Figure 3 A crystallisation system in solid, liquid and gaseous states

Figure 3 shows an emergent model of crystallisation of material [Jankovic and Dumpleton, 2000], which first started from random positions for all crystals and self-organised its behaviour towards a fully crystallised state. After the initial crystallisation is completed, adding heat to the model reverses the process, taking it into liquid and gaseous state. Similarly to the boids model, the crystals are driven by a small number of rules, based on the Coulomb's law for electrostatically charged particles. The simplicity of the component models is in sharp contrast with the complexity of the system model.

Other models of complex behaviour, such as queuing of traffic, chemical reactions, and others were developed on the basis of the same principles: a small number of rules acting on a component level, and a complex system behaviour emerges from the bottom-up, without explicit programming.

The question is then can the same be done for structures?

4. BOTTOM-UP STRUCTURES

Learning from the experience of models of complex systems described in the previous section, it appears that complex behaviour of structures could arise in a similar way: small number of rules for each component, in this case Newton's laws of motion, and the components are made to interact. We now investigate whether this approach would create analogue models of structures capable of dynamic simulation of their behaviour, without explicit programming, but rather with self-organisation of the components into emergent system models that build themselves. In particular, we examine whether behavioural models of animal movement, such as boids, could be applied to structures.

4. 1 Transfer Of Behavioural Modelling Principles To Structures

In an attempt to transfer behavioural modelling principles to structures, we now examine and compare the essence of both.

From the analysis of existing behavioural models it was found that their essence could be described as follows:

1. Multiple components
2. Simple rules for individual components
3. Local interaction between the components
4. Global model emerges from the local interaction without explicit programming

Similarly, the essence of structures can be described as follows:

1. Multiple components
2. Simple rules for individual components: Newton's laws of motion
3. Local interaction between the components

Thus the first three aspects of behavioural models are also present in structures. The question is whether the fourth aspect would also occur, i.e. whether the global model of a structure would emerge from the local interaction of the components without explicit programming, and therefore, whether or not the emergent model can be created in this way. The answer to this question will be given by experimental research reported below, which involves development, testing, and validation of models of structures created on this basis.

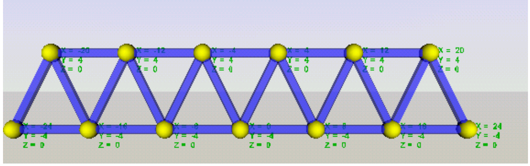


Fig. 4 Bottom-up model of a two dimensional truss

4. 2 Development And Testing Of Dynamic Simulation Models

```
#Component prototype
PROTO Component [
  #PROTOtype declaration
  eventIn SFSomething protoEventIn
  eventOut SFSomething protoEventOut
]
{
#PROTOtype definition
  DEF protoNode someNode {
    DEF someSubNode someOtherNode {}
  }
#The script inside the prototype is
mapped
#to PROTOtype inputs, outputs, and
internal #nodes using IS and USE

 Script {
  eventIn SFSomething scriptEventIn IS
protoEventIn
  eventOut SFSomethingscriptEventOut IS
protoEventOut
  field SFNode scriptNode USE protoNode
    url "javascript:
         scriptNode.someSubNode =
       someValue;
    "
 }
}
```

Fig. 5 VRML and JavaScript pseudo code for component model architecture

In order to answer the question of whether or not the boids model is applicable to structures, we will first develop an emergent model of several typical structures. This section reports on the development of two types of emergent models of structures: interconnected bars, such as trusses, and those made of continuous materials, such as beams.

4.3 Development Of Emergent Models Of Trusses

4.3.1 Model Description

A model of a simple triangle truss was developed using VRML97 and JavaScript, and consisting of joints and bars as main components. Following the principles of structural dynamics, the assumption was made that the mass of the truss was concentrated only in the joints. The joints and the bars were modelled as separate, independent components, each considered as an object in terminology of the object-oriented programming. The objects were connected through inputs and outputs only, in a similar way in which they are connected in reality. The bars were modelled as weightless springs. A viscous damping mechanism was adopted, and the damping force modelled to act in the opposite direction to the joint's velocity. Each component was defined on the basis of the Newton's laws of motion. The model of the triangle truss was then scaled up into a model of a simple truss (Fig. 4).

4.3.2 Component Model Architecture

The component architecture involved the use of PROTO(types) and Scripts (Fig. 5). The component PROTO was responsible for dealing with external components and the Script inside the PROTO was responsible for implementation of simple rules that determined the component behaviour.

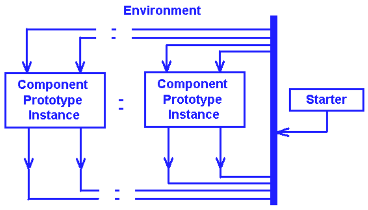


Fig. 6 System Model Architecture

The parameters and variables from the PROTO interface declaration were mapped to the Script parameters and variables. This enabled the prototype inputs to access the Script inside the PROTO directly.

All components of the emergent model were created by a process of instantiation (creation of working copies) of the same PROTO.

To ensure the compatibility of inputs and outputs so that outputs of one instance of the PROTO could be connected to inputs of another instance of the same PROTO, the PROTO's inputs and outputs were chosen carefully to be of matching data types.

The Script inside the PROTO was mapped to its geometric properties in order to effect the visual representation of behaviour of individual instances. This created a generic component model suitable for interaction with other components.

4.3.3 System Model Architecture

It was found that complex interactions between system components are best handled using a separate "container" PROTO(type) named Environment, capable of dealing with the influences between a large number of pairs of components (Fig. 6, Fig. 7).

```
#Environment
Transform{
  #instantiation of Components
         children [
      DEF Component1 Component{
   #instantiation parameters here
   }

   DEF Component2 Component{
   #instantiation parameters here
   }
       ]

DEF Starter TimeSensor {}
#Routing connects system components
#and system starter
   ROUTE Starter.time TO
      Component1.someEventIn
   ROUTE Component1.someEventOut TO
      Component2.someEventIn
   ...........
   ROUTE ComponentN.someEventOut TO
      Component1.someEventIn
}
```

Fig. 7 VRML and JavaScript pseudo code for system model architecture

4.4 Testing Of Emergent Models Of Trusses

Figure 8 shows an example of the same two dimensional truss as in Fig. 4, but after the application of load.

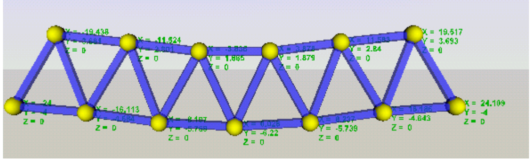


Fig. 8 Bottom-up model of a two dimensional truss after application of load

It was found that the model behaved realistically, and returned to the original state after being disturbed from the equilibrium. The co-ordinates on the side of the joints were changing as the system went from the initial state, to the maximum displacement state, and back to the equilibrium. The user was able to interact with this model in the real time, and examine its behaviour.

4.5 Development Of Emergent Models Of Beams And Portal Frames

Modelling of beams and portal frames that are made out of continuous material, required a different approach, in comparison with the approach used for modelling of trusses. Unlike the truss models, in which all of the individual components were manipulated by the environment, the beam models had to contain this capability within the component itself. Therefore, the model had to be completely redesigned, in order to satisfy this requirement. The very first exploratory model behaved as a lump of jelly, and was named "jelly-box". This model was subsequently scaled up and extended into a "jelly-beam" and "jelly-portal frame", all of which are described below.

```
for (j=0;j<NumberOfPoints;j++) {
  for (k=0;k<NumberOfPoints;k++) {
    if(j != k)
      force[j]+=
        displacement[j]*elasticity-
        velocity[j]*damping;
      acceleration[j]=force[j]/
        pointMass;
      velocity[j] = acceleration[j] *
        timestep;
      position[j] += velocity[j] *
        timestep;
    }
}
```

Fig. 9 Pseudo code of application of Newton's Laws to interaction of jelly-box corners.

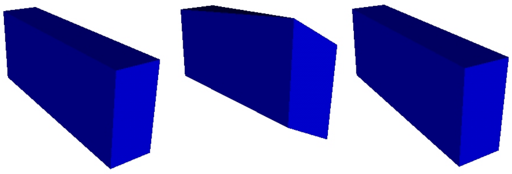


Fig. 10 "Jelly-box" before, during, and after the application of load

4.5.1 Jelly-box

Jelly-box model was based on a box looking shape that had a point mass in each corner. Each corner was then made to interact with each other corner on the basis of Newton's laws, through elastic forces and friction/damping forces (Fig. 9). This gave rise to an emergent model of an elastic box, which enabled the user interaction in the real time (Fig 10).

4.5.2 Jelly-beam

Scaling up of the jelly-box model, by adding multiple segments of the elementary component, created a jelly beam model. The code that runs the jelly-beam model is essentially the same as in Fig. 9, the only difference being the number of points that describe the shape of the beam. The operation of the model, with load applied interactively by clicking and dragging one of the corners is shown in Fig. 11.

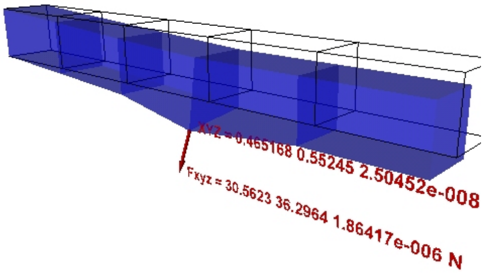


Fig. 11 User interaction with a Jelly-beam model

4.5.3 Jelly-portal frame

Further scaling-up of the jelly-beam model, by adding more points and introducing corner components, created a jelly-portal frame model. The operation of this model, loaded with a constant force, is shown in Fig. 12. This model, like the previous two, was also capable of real time user interaction.

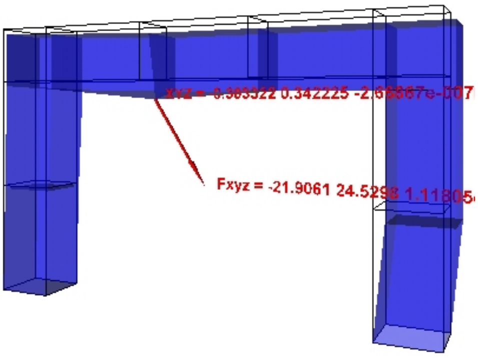


Fig. 12 User interaction with a Jelly-portal frame model

All of the three models described in this section: jelly-box, jelly-beam, and jelly-portal frame, were created using the same extendable code. Consequently, any new points added to create new shapes are automatically assigned physical properties. The resultant interaction between the points self-organises the model and gives rise to its emergent behaviour.

All of the emergent system models described here were created spontaneously, without explicit programming. Effectively, these models built themselves. In the next section we evaluate their performance.

5. EVALUATION AND VALIDATION

The emergent models of structures reported here have integrated the calculation and visualisation into one, and thus enabled the user to interact dynamically with the models. The models have been developed, using interpreted languages VRML and JavaScript, and low specification computer hardware (300 MHz Pentium II).

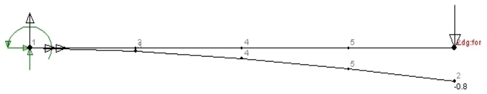


Fig. 13 Reference case for comparison obtained from LUSAS

However, before engineers can adopt these models as simulation design tools, the question that needs to be answered is whether they are as accurate as conventional models.

5.1 Validation Of Accuracy

Validation of the emergent models is the final stage of the model development in which the accuracy of the model is investigated by comparing it with validated conventional models.

The conventional method chosen for comparison was LUSAS [FEA Ltd. 1999], a software package running the standard Finite Element Method. The LUSAS model of the beam was created using four main horizontal segments, with overall dimensions of the beam of 10 m length by 1 m height (Fig. 13).

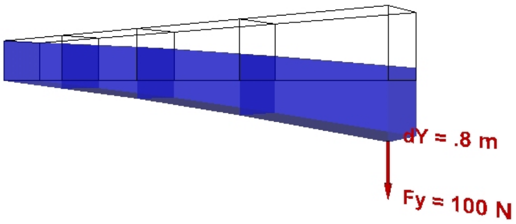


Fig. 14 Jelly-beam model responding to a fixed force and giving the resultant displacement

This was compared with the jelly-beam model, which had the same dimensions, and a capability to operate in automatic mode in response to a fixed force, giving the resultant displacement (Fig. 14). The stiffness of material was calibrated to achieve the convergence of results.

The comparison of results of operation of jelly-beam emergent model (EMERGENT), LUSAS model of the beam, and a theoretical beam model is summarised in Table 1. Although the three models give identical results, suggesting that the jelly-beam emergent model is accurate, this is not quite the case with all points along the length of the beam, suggesting that further fine-tuning of the model is needed.

	F = 10 N	F = 50 N	F = 100 N
Beam theory	0.080	0.400	0.800
LUSAS	0.080	0.400	0.800
EMERGENT	0.080	0.400	0.800

Table 1 Comparison of displacements in metres of three different beam models

5.2 Features Of The Emergent Models

It has therefore been shown that the emergent models reported here integrate physics and geometry into analogue models of structures. While responding dynamically to user interaction, the emergent models can provide to the user information on the values of parameters inside the model, and can therefore potentially replace the need for conventional design calculations. The accuracy of the models appears to be consistent with the conventional methods, and at the same time the models appear to self-organise through the interaction of the model components, without explicit programming. The latter point is particularly significant for modelling of structures that were not considered during the initial development of the method [Jankovic, 2000]. Another significant capability of the emergent models is the capability to simulate both static and dynamic behaviour of structures.

5.3 Limitations Of The Emergent Models

The emergent models use discrete time steps as the basis of its operation. This can potentially cause instability of models, if the time step required to complete necessary calculations is greater than the achievable frame rate. As reported by [Jankovic and Dumpleton, 2000], this limitation occurs in other emergent models as well, and indeed in any other model where continuous time from the real world is replaced by discrete time in the computer model. This can be prevented by reducing the time step, at the expense of speed of performance of the model.

Another limitation of this method, and of emergent models in general, is the need for the program designer to set a number of model parameters. In the emergent models described here, the stiffness of material had to be varied in several steps until the results converged with the conventional methods. The emergent models therefore need to be calibrated before use.

6. CONCLUSIONS

The starting premise for this work was the need for better design tools for structural engineers. It was found that the main reason why visualisation could not be integrated into conventional calculations was the top-down approach using the finite element method.

To overcome the limitations of the conventional top-down approach, the solution was sought in the field of complexity, and in particular, in the behavioural models of animal movement, such as boids. Fundamental similarities between boids and structures were found, and the model architecture of boids was successfully transferred to structures. Two types of structures were modelled on this basis: interconnected bars (trusses) and continuous materials (beams).

The new models were capable of dynamic response to user interaction, and therefore satisfied criteria for dynamic simulation models. The emergent model of the beam was validated by calibration of its parameters and comparison with conventional and theoretical models, showing general convergence.

Two main limitations of emergent models were identified. The first limitation was the use of discrete time to approximate continuous time, which can potentially make the models unstable under certain circumstances, but can be alleviated by reducing the time step. The second limitation was identified as the need for the program designer to set a number of parameters before the simulation. However, as one-off activity, this was not considered to be a significant problem.

The most important feature of the emergent models is self-organisation of the system model of the structure through interaction of its components, without explicit programming. This makes the emergent models flexible, and suitable for different forms of structures, whilst providing access to internal parameters of component models that are necessary for structural design. By self-organising, these models have effectively built themselves.

7. REFERENCES

Connell, M. and O. Tullberg, 2000. "A Framework For The Interactive Investigation Of Finite Element Simulations If Virtual Environments". In Proc. of *Developments In Engineering Computational Technology*, Civil-Comp Press.

Jankovic, S. et al., 2000. "Bottom-Up Virtual Reality Model Of A Shallow Two-Bar Truss With Snap-Through Behaviour." In *Proceedings of ACME2000*, London 16-19 April 2000.

Jankovic, L. & J. Dumpleton, 2000. "Emergent Modelling Of Complex Systems In VRML". In *Proceedings Of Eurographics UK 2000*, Swansea 4 -6 April 2000.

Kauffman, S., 1996. *At Home In The Universe - The Search For Laws Of Complexity*. Penguin Books.

FEA Ltd, 1999. *LUSAS Examples.* Version 13.2-0.

Langton, C. G., 1992. Life At The Edge Of Chaos. Proceedings Of Artificial Life Ii. Addison-Wesley, 1992.

Reynolds, C. W., 1987. "Flocks, Herds, And Schools: A Distributed Behavioural Model". *Computer Graphics*. Vol. 21, No. 4.

ACKNOWLEDGEMENTS

This research is collaboratively funded by the UK EPSRC grant No. GR/M75273 and by the following industrial partners: WS Atkins Consultants Ltd, Oscar Faber Group Ltd, Maunsell Ltd, Halcrow Group Ltd, Mott MacDonald Ltd, Hyder Consulting Ltd, Ove Arup and Partners, Kvaerner Technology Ltd, R O'Rourke & Son Ltd, and InteSys Ltd.

THE AUTHORS

Dr L. Jankovic Since his PhD graduation from the department of Mechanical Engineering, University of Birmingham in 1988, Dr Jankovic lectured at the Universities of Central England and Birmingham. He is a founding director of InteSys Ltd, a Fellow of the Institution of Analysts and Programmers, and a registered Chartered Engineer. He worked for European Commission as expert evaluator of research proposals, and co-ordinated several international research projects. He also worked as a consultant for a number of commercial organisations.

His main research interest is in modelling and simulation of complex dynamic systems, including time series, structures, and artificial life systems. He patented a microchip for modelling and simulation of time series, for which he won two Smart Awards for Research and Technology from the UK Department of Trade and Industry.

Mr S. Jankovic After graduating in Faculty of Civil Engineering, University of Belgrade, Mr Jankovic worked in industry, first on design of two large steel bridges and after that as an internal quality control engineer at bridge construction sites. He joined School of Civil Engineering, University of Birmingham in 1999 as a postgraduate student and become Research Associate in a project entitled "Application of Bottom-up Virtual Reality Modelling to Optimisation of Structural Design in Construction Industry". His research covers modelling of structural behaviour in virtual reality, for which he has developed a number of analogue interactive virtual reality models of structures.

Dr A. H. C. Chan graduated with first class Honour in BSc (Eng) degree from the Department of Civil Engineering, University of Hong Kong. He continued with MPhil degree in the same department then obtained his PhD degree from the University College of Swansea, Wales. He was employed as Research Assistant at Cambridge University Engineering Department and Lecturer at University of Glasgow. He is now Reader in Computational Engineering, School of Civil Engineering and the head of the Engineering Computation Network from August 2001, School of Engineering, University of Birmingham. His main interest is in numerical methods and computer methods for Civil Engineering applications. He is a chartered engineer and a member of British Computer Society.

Dr G. H. Little After graduating in Mechanical Sciences at Cambridge University, Dr. Little joined Freeman Fox and Partners as an Assistant Engineer, and worked on the design of several big steel bridges, including the first Bosporus bridge. He returned to Cambridge as a research student, to study problems related to the steel box girder research effort of that time. His PhD thesis was concerned with the collapse behaviour of steel columns and stiffened plates. In January 1974 he joined the School of Civil Engineering, The University of Birmingham, as a Lecturer, becoming Senior Lecturer in 1991. Dr. Little's research activities cover the behaviour and design of metal structures, and composite structures.