Empirical Modelling of Products

Carlos N. Fischer†, Meurig Beynon*

†Departamento de Estatistica, Matematica Aplicada e Computacional,
Universidade Estadual Paulista - UNESP, Brazil.
***Department of Computer Science, University of Warwick, UK.**
cfischer@rc.unesp.br, wmb@dcs.warwick.ac.uk

Keywords: Product modelling, product design, computer-based model, empirical modelling, computer simulation.

Abstract

Modelling – the activity of creating a representation of a physical object – has a very important role in modern product design. Modelling has benefited greatly from the use of a computational environment to construct models. We present here the concepts of an approach known as Empirical Modelling (EM) that offers an environment with a high level of interaction with a computer-based model. This environment allows the construction of models with a richness of interaction, openness and flexibility. Two case studies, simulating real products, are used to show how the interesting principles of EM can be applied in the development of tools to support product modelling. The application of EM in an educational environment is also discussed.

1. INTRODUCTION

The area of product design has received significant benefits from the use of computer-based models. An appropriate computational environment for creating models must provide the designer with a high degree of interaction with the model. Such an environment should help to make the development of a new product, or the modification of an existing one, easier and quicker [8,15]. It should also be helpful for an engineer who wishes to explain either the characteristics of a product or particular design decisions taken during its development.

In modelling a product, the interactions of the modeller should ideally support the constant revision that is needed to obtain the best representation of the product. The knowledge acquired from such interaction is very helpful in suggesting improvements to the product itself.

Empirical Modelling (EM) – so called because the modelling principles are based on observation and experiment – is an unconventional approach to modelling systems under development at the University of Warwick [13]. It offers an environment very appropriate for applications that demand a high level of interaction with the computer-based model. Central to the Empirical Modelling perspective is an emphasis on the power of the computer to represent states, in particular, states which are easily interpretable. An EM model metaphorically represents a particular state of the system under study.

In this paper, section 2 presents a brief overview about product modelling and relevant topics. Section 3 introduces the concepts of the EM approach, describing some interesting characteristics and tools. Two case studies, simulating a digital watch and an IDE hard disk controller, are detailed in section 4; the objective is to show the application of the principles and tools of Empirical Modelling in the construction of models, and demonstrate its merits as an alternative approach to developing tools for product modelling. Finally, section 5 describes some aspects of the applicability of Empirical Modelling in an educational environment.

2. THE PRODUCT MODELLING PROCESS

The design process is the process of generating a solution that meets the requirements imposed on a product [9,15]. The conception of the design process proceeds in parallel with current product knowledge evolution: as the design process is conceived, the knowledge of the designer about the current product evolves. Moreover, the evolution of knowledge about the product leads directly to the support of the design process. Two important objectives in developing a design process are optimising the activities for achieving the design requirements and creating alternative solutions that allow the construction of several variants of the product.

Product modelling is the process of creating a concrete or virtual representation (a model) of the characteristics of a physical product under study. Modelling is a very important activity in modern product design processes [8,9]. An appropriate model provides the designer with a richer, higher level and more intelligible design concept than the underlying natural language. Using such a model reduces ambiguity, makes it easier to check for incompleteness and may improve understanding about the object. Due to the increasing product complexity and variety, complex

functional requirements, rapid product changes and the increasing time constraints on design processes [9], product modelling has become more and more important in the design process. Through modelling, the designer can create a new product, modify an existing one, and check and test features of the product. Product modelling research focuses on constructing tools to support the efficient construction of appropriate models that help to meet the objectives of the design process.

The computer is already extensively used as a tool for modelling within product design. It must be recognised that the entire design process contains a uniquely human element: creativity. Thus, the computational environment for modelling must provide appropriate tools through which the modeller can exercise his creativity. Such an environment must also offer the modeller a high level of interaction with the model and provide tools for modifying the model in a simple and quick manner, aiming at achieving the design requirements and opening the possibility of creating alternative solutions for the product. In many cases, another important feature is that the modelling environment should support animation of the model. Ideally, a modelling environment should also enable designers to share knowledge about the product and integrate their ideas about its model.

In this paper, a computer-based model is referred to as an *artefact* and the object under study as the *referent*.

3. THE EMPIRICAL MODELLING APPROACH

Empirical Modelling (EM) is a collection of principles and techniques that has been under development at the University of Warwick since 1983 [2,3,13]. EM offers a set of tools for modelling and visualisation of general systems that allows a user to build an artefact that is always open to subsequent extension, refinement and revision. Animation of an artefact is also possible, through visualisation of sequences of states of the referent subject to some behaviour.

The EM principles are based on three fundamental concepts: *observable*, *dependency* and *agency*. The term *observable* is used to refer to elements that relate to our understanding of the referent. A *dependency* is a relationship among observables that expresses expectations about how the values of observables are indivisibly linked in change. An *agent*, with its associated observables, has privileges for actions to change the values of those observables. For example, when modelling a room, the modeller might consider the light and the light switch as observables that can be on or off; might state a dependency between the light switch and the light: when the light switch is pressed on, the light comes on at the same time; and, an agent could be a user of the room who may switch on and off the light switch.

The identification of the relevant properties (observables, dependencies and agents) of a referent is subjective, based on the observation and interpretation of the modeller. Generally, this identification is provisional as it depends on the modeller's personal experience of both the referent and the artefact. Through experimentation with the artefact, the designer gains knowledge and new ideas related to the referent, so new properties may arise apart from those initially identified. In an EM artefact, any new property can be added to the artefact at any stage during the modelling activity without the need to revise the whole artefact or revert to an earlier version. Moreover, as the changes in observables can be executed during the simulation, modifications in the artefact are immediately executed, which allows subsequent revision of the artefact and the complete development of the artefact in an incremental fashion. This flexibility of EM is a distinct concept in modelling from many conventional modelling approaches in which the modeller has to preconceive what the inputs and outputs are going to be before starting the construction of the model in order to prescribe the required behaviour. That is, the boundary of the proposed system must be determined in advance. If there is a need for a new input or output, e.g. an additional system feature is required, then the whole system may have to be revised and re-designed at substantial cost. Part of the EM flexibility is that in EM the focus is initially on the *state* of a model or domain, rather than on the behaviours the modeller wishes to produce.

The dependencies among observables are represented by *definitions*. A typical definition takes the form $x = f(y,z)$ where x, y and z are artefact observables associated with the referent observables, and f is a function that defines the relationship between those observables. Such a definition expresses a dependency relating x, y and z that is automatically maintained by the EM tools. That is, changes in the observables y or z will result indivisibly in a change in x, the dependent observable. Such dependency amongst observables obviates the need to verify whether a dependent observable is up to date in the artefact. A definition can also assign an explicit value, as in $y = 5$. The set of definitions forms a *definitive (definition-based) script* to be interpreted by the EM tools.

The particular values of the observables and the dependencies associated with the artefact represent one of the possible states of the referent. Changes of state occur in the artefact either through the re-definition of observables or the addition of new definitions. Through automatic re-definition of observables and consequent visualisation of a sequence of representative states of a referent, an EM artefact can be animated so as to give the impression of a behaviour.

Another important characteristic of EM is that there is a distributed version of the modelling environment [18] that allows communication between users who can access and visualise their representations of the same artefact over a network. In this way, many designers may interact with an artefact, allowing their viewpoints to be integrated during the modelling process.

The principal modelling tool that has been developed at Warwick is the TkEden interpreter, which is implemented in C. TkEden supports definitive scripts in which the variables represent a variety of two-dimensional graphical elements, including shapes, point and lines, text strings, windows and displays. TkEden incorporates three definitive notations: DoNaLD, for line drawing, SCOUT, for screen layout, and EDEN, an evaluator for definitive notations that allows definitive scripts to be formulated over scalar types, non-homogeneous recursive lists and strings.

The EM tools are still primarily research ones, but they are sufficient for proof-of-concept of the principles of EM, and previous work [5,7,16] has demonstrated that they allow the construction of artefacts that are always open for interaction and revision. Moreover, any components from an artefact can be integrated within another artefact, thereby making modelling easier and quicker.

4. EM ARTEFACTS AND PRODUCT MODELLING

The qualities of Empirical Modelling (EM) described in section 3 are particularly significant when the modeller needs to construct an artefact with a high degree of interaction and flexibility, as in product modelling. To illustrate this, two case studies in artefact development are presented in this section: the *Digital Watch* and the *Disk Controller* artefacts. The emphasis is on the nature of the model development process and the application of the EM principles, not on any details of the product functionality. The objective is to show that EM offers an open-ended environment that provides an alternative approach to product modelling.

4.1 The Digital Watch Artefact

The first case study is a simulation of a watch. The watch artefact, with its digital and analogue displays, is depicted in Figure 1. The artefact reflects the watch in its many different display modes: presenting the current time, date, alarm time and chime time, or serving as a stopwatch. These functions are realised on the digital display that simulates six LCDs. Six coloured input buttons located in the digital display are used to set the power on, change the display mode and update the time and date. The artefact includes a statechart, originally presented in [12], that shows the current mode of display of the digital watch and how the display functions of the watch are affected by button presses. Each node in the statechart represents a display mode (the current display mode is highlighted in the statechart) and each transition between display modes is represented by an edge that is coloured according to which input button is pressed. The statechart is viewed as a representation of reliable knowledge about how the watch will react to inputs. All the changes of state within the statechart represent particular observed behaviours.

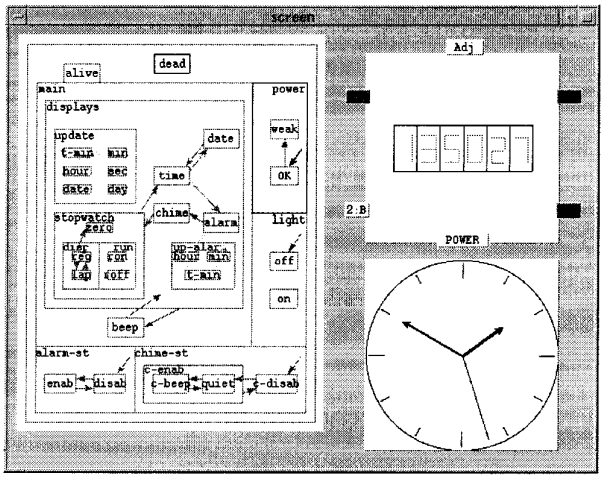

Figure 1. EM artefact of a Digital Watch.

The artefact represents a particular state of the watch, together with latent state transitions that reflect what the modeller expects to happen in response to a particular action. Each state of the referent is specified in terms of the current values of a collection of observables of the artefact. The visual elements in the model represent the current values of the observables. These values can be changed by external and internal agents of the artefact. An example of an external agent is a human that can set the power on by pressing the *Power* input button; this action changes the value of an observable called *power_s* to a non-zero value that triggers a procedural action for switching the watch on. A similar action continuously checks the current time in the artefact and after a period of time changes the value of *power_s* to zero (simulating exhaustion of the battery) and so stops the watch. That procedural action is an example of an internal agent of the artefact. Another internal agent continually updates the value of an observable called *new_time* with reference to the computer system clock; each new value of *new_time* changes the state of the artefact. These consecutive state changes are automatically displayed on the screen, and so simulate the watch in operation.

The artefact associates states of a real watch with instantiations of agents. For example, the *watch* agent represents the watch itself and the *power* agent its power

supply. In the specification of each agent, the identifiers refer to observables that are significant for the agent. In the model, the *power_s* observable represents the energy that is supplied to the watch and an observable called *live* indicates whether the agent *power* is present or absent:

$$live = (power\_s > 0).$$

The dependencies in the artefact reflect several different types of agency. The primary purpose of these dependencies is to bind the visual elements and the indivisible internal state of the model together so to reflect interaction with the external observables they represent. The model provides a simple interface through which the modeller can simulate button pressing. This interface gives scope to extend and modify all the definitions, functions and triggered actions interactively. Changing the current time affects the displays on the digital and analogue watches indivisibly. Moving the minute hand of the analogue clock moves the hour hand. Changing the mode of display indivisibly affects the highlighted state in the statechart and the watch display.

Simple instances of definitions and agencies are illustrated by the analogue clock. In an engineering model, the hands of the clock may be coupled mechanically so that when the minute hand moves, the hour hand moves simultaneously. In the artefact, this dependency can be expressed by an explicit definition:

$$angle\_hour\_hand = (angle\_min\_hand / 2\,\pi) * (\pi / 6).$$

Whilst such a definition is extant in the artefact, any movement of the minute hand entails simultaneous movement of the hour hand, both in simulating the clock in its normal operation and in setting the clock. The normal operation of the clock is associated with agents that can act autonomously without the intervention of the designer. The designer can nonetheless intervene to interfere with these agents, for instance to simulate a clock running down or malfunctioning. The role of the user in setting the clock, which conforms to no preconceived pattern of interaction, can be played directly by the designer.

The relation between second and minute hands illustrates another role for agency in constructing an artefact. In our chosen clock mechanism, the motions of the second and minute hands are mostly independent but are synchronised by updating the position of the minute hand at regular intervals. In this case, the movement of the minute hand is not within the scope of the indivisible action of moving the second hand. To model this, an independent agent was introduced. This monitors the position of the second hand and updates the minute hand in discrete steps as the second hand registers that a minute has elapsed.

The development history of the watch artefact shows that EM artefacts are always open to revision and extension. Three people have contributed to building the watch artefact in its present form. The basic statechart and digital display were developed by the second author in 1992, details of the statechart, the buttons for changing the display mode, and the analogue clock were subsequently added by his research student in 1994, and the functionality of the watch was completed by the first author in 1999, who added new definitions and procedural actions to the model, and buttons for updating time and date and for switching the clock on to the interface. Interaction with the evolving watch artefact played a far more significant role in the modelling activity than communication between the contributors. Indeed, the artefact itself has proved to be a more useful repository of knowledge about its construction than memories of previous modellers.

The building of the watch illustrates some characteristic features of EM. The artefact was incrementally constructed whilst running as a background process on a workstation. Control over dependency and agency made it possible to take account of the new views of the design process as they arose. Interactive development sessions typically involved the addition of a group of definitions to attach a new visual component to the artefact or of procedural actions to simulate the introduction of new agents. It was occasionally necessary to reload the current files of definitions and actions (for instance, in the event of bugs in the interpreter, or system maintenance beyond our control), but several days typically elapsed between such events. The development exercise in some ways resembled conventional engineering: it was sometimes useful to develop portions of the script independently, or to trace problems by extracting pieces of the script and exercising them in isolation. The ease with which we could adapt alternative computer models on the fly contrasts with the considerable ingenuity that would be required to modify a clock mechanism.

4.2 The Disk Controller Artefact

The artefact of the second case study is shown in Figure 2. This artefact is intended to simulate an IDE Hard Disk Controller [1] taking into account the more relevant characteristics of this type of digital subsystem. A real IDE Controller is a subsystem to control hard disks in a computer system, transferring data between a disk and other parts of the computer. It is basically composed of a set of registers, buses and hardware that manages all the subsystem. The registers are used to specify the type of action to be executed on a disk and, if data transfer is involved, where to read or write data on the disk.

In the artefact, there are four display windows. The top left window represents the main components of a host computer system. The bottom left window features a set of input buttons used to read and write to the IDE Controller registers. The other two windows depict the read/write heads and arms of two hard disks. The two most recently accessed cylinders can be seen in each disk. The artefact

simulates some operations performed by the physical IDE Controller; for example, reading of a register and data transfer between a disk and the DMA device. For each operation, the first window in the artefact highlights which parts of the computer and which components of the IDE Controller are involved in that operation.

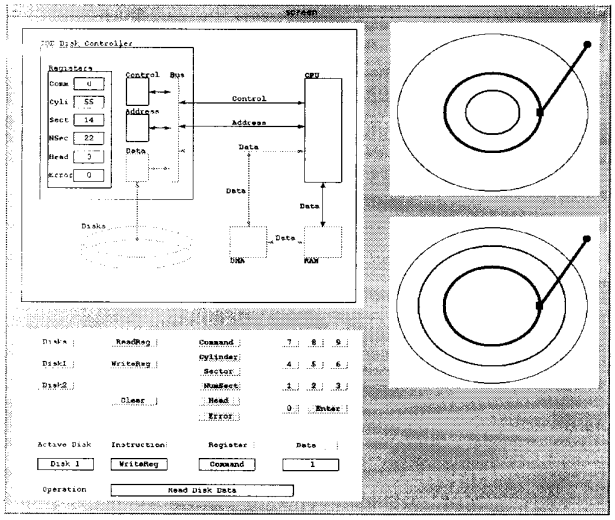

Figure 2. EM artefact of a Disk Controller.

As explained above, the current values of the observables of an EM artefact represent a particular state of the referent. Changes in the values of these observables imply new representative states for the artefact, which are reflected in the visual elements of the model. For example, when the action of moving the read/write heads of a disk is required in an operation, an observable called *jump* is updated several times (from the initial to the final disk cylinder values involved in the action). Each *jump* updating operation changes the value of another observable, *poshead*, that determines the position of the disk heads. Each new value of *poshead* signifies a new representative state for the artefact that is automatically visualised on the screen. These changes in the state of the artefact are animated to imitate the movements of the disk heads.

The construction of the Disk Controller artefact illustrates some interesting characteristics of EM. The artefact was developed by the first author in a single month with no prior experience of EM other than his brief work on the Digital Watch artefact. The artefact was constructed stage-by-stage: first the computer system structure; then the input buttons; and finally the disk representations. Several types of re-use featured in the modelling activity. Some parts of the Digital Watch artefact were, after suitable adaptation, integrated into the new artefact. For example, the statechart was "transformed" into the window that represents the main components of a computer system, and the codes for the structures of the coloured input buttons of the watch were simply replicated (with new texts and positions only) to construct the set of buttons for reading and writing the IDE controller registers. The first implemented version of this artefact included only one disk. Through direct replication of pieces of script (for disk drawing and procedural actions) it was very easy to introduce another disk to construct the artefact presented in this paper. The disks of this version are independently operated; an observable called *disk_s* specifies the disk being considered. However, with a very simple modification in the script, it is possible to make the disks work together, introducing a single dependency to simulate a scheme for automatic disk backup, for example. Similar code replication could be used to simulate more sophisticated techniques for data storage using several hard disks, such as in the RAID (Redundant Array of Inexpensive Disks) technology [10,11]; such an artefact could be used to explain how this type of technology, or a possible new disk architecture, works.

4.3 Product Modelling Using EM

The case studies presented above demonstrate the power of the Empirical Modelling (EM) principles and tools to construct artefacts that are always open for interaction and revision. The EM environment offers a high level of interaction with the model, allowing the modeller to revise the model continually in order to obtain the best representation of the product. The EM features of immediate visualisation of artefact modifications and model animation are very important for increasing the knowledge of the modeller about the product. Through interaction with the artefact, the modeller acquires product knowledge, which leads to the evolution of the modelling process.

The openness of the EM environment allows components from an artefact to be integrated into another artefact, as shown in the case studies, making the modelling process easier and faster. Using this facility of EM, more complex components of a product can be separately modelled and then joined within a unique artefact. Those separate models could be made by modellers working in a distributed environment for EM. The modifications in an EM artefact can be made at any stage during the modelling process, without the need to re-start the process, allowing the complete development of the artefact in an incremental fashion. In this way, it is easy to build one artefact from another and in particular create several versions to an artefact to represent alternative designs for the same product.

The dependency between observables in an EM artefact can be exploited to maintain relationship amongst positions,

proportions and shapes of components of the artefact. For example, in the Disk Controller artefact, the action buttons "0" to "9" and "Enter" are positioned with reference to the position of "0" button. When the position of "0" button changes, the positions of the other ten buttons change as well, due to the automatic updating that is executed in dependent observables. In the Digital Watch artefact, the first LCD display was defined as a square shape and the rest as translations of that LCD. If the shape of the first LCD were to be changed (for example, to a circle or ellipse), the shape of the other five would change as well.

The case studies described demonstrate the potential of EM as an alternative and appropriate approach in the development of tools aimed at product modelling. To realise this potential more fully, some limitations of the current EM tools need to be addressed. These include:

- limited graphical capability: the TkEden environment does not support the high quality 3D modelling and rendering generally needed for the presentation of the product model;
- no end-user interface: TkEden does not provide a suitable environment for the end-user;
- restrictions on scale: an expressive model can become very large; even existing TkEden scripts can have thousands of definitions, and cannot be interpreted efficiently on modest workstations. Realistic product models would be an order of magnitude larger.

Several developments promise improved capacity and efficiency of the EM environment for modelling of systems in general [2]. Recent research has been directed at integrating EM tools with 3D modelling packages, such as OpenGL and Hyperfun [14], and at implementing higher-order definitions that assist the replication of patterns of observables, dependency and agency. Improved management of patterns has important applications in the field of product modelling. For example, in the current implementation, all action buttons are square; in the future, designers could be given the possibility of choosing between several action button shapes or creating their own pattern. Due to the increasing time constraints on product design processes, the performance of the modelling system is an important issue; although the use of higher-order definitions cited above can diminish the size of the artefact, another concern is the amount of dependency maintenance performed during a modelling process. Promising approaches to more efficient dependency maintenance currently under investigation include the development of new architectures in which dependency is maintained close to the machine code level, and optimised parallel algorithms for storing and updating dependencies.

5. EM IN EDUCATIONAL ENVIRONMENT

EM is an approach to modelling that emphasises observation, experimentation and interaction with the models throughout their development process. The modelling activity in EM is associated with empirical knowledge about the referent. Empirical knowledge, unlike theoretical knowledge, is always open to subsequent extension, refinement and revision [6]. This means that an EM artefact can be considered as essentially incomplete, and must be complemented by interaction. Thus, EM models can be exploited as interactive environments for learning and training processes [4] - a theme developed at length in a companion paper [17].

The EM environment can provide the user with a high degree of interaction with an artefact that can inform the user's observation and experiment with the referent. EM is concerned with the empirical process through which knowledge and understanding are acquired during the construction of the model: through observation and experimentation with the model, the user acquires insight about the referent. This insight can then be used to extend the artefact. Once again, new knowledge and understanding is obtained that can then be used to revise and extend the model. Thus, knowledge about the referent evolves as the modelling progresses.

The way in which a user's knowledge about some subject evolves can be monitored in an EM environment so as to evaluate the learning process. An EM model can be the platform for activity to be carried out by a user that involves executing specific tasks that complement the model. The user's level of comprehension and how much the user is learning about each task can then be evaluated by observing whether or not the user is complementing the artefact in an appropriate way.

Though the principles for the development of educational environments from EM models are best illustrated with reference to models that have been constructed for a specific pedagogical purpose [4,5,6], the case studies presented here exhibit certain common general principles concerning interaction and learning. Both cases demonstrate how understanding originates with developing expectations about the response to interaction. This can take the form of local knowledge about the effect of a single action in a particular state, but can also involve developing insight into various relationships between observables with respect to a family of possible sequences of interactions. For the experienced user, in the role of a teacher, interaction with the model is helpful as a way of assessing the current status of the model, and confirming the viability of a particular strategy for further development. For the inexperienced user, in the role of a student, interaction is useful as a way of checking comprehension and resolving obscure or uncertain issues.

6. CONCLUSION

The Empirical Modelling (EM) approach described in this paper is based on a view of computation and programming that is significantly broader than conventional views. By invoking the concepts of observable, dependency and agency, EM aims to build computer-based models of products that reflect our conceptual models of products. The model development exercise in some ways resembles conventional engineering: it is sometimes useful to develop portions of the script independently, or to trace problems by extracting pieces of the script and exercising them in isolation. The ease with which we can adapt alternative computer models on the fly contrasts with the considerable ingenuity that would be required to modify a physical engineering prototype. EM can offer both cognitive and operational support to product modelling from the very early, conceptual stages of modelling. The potential benefits of applying EM to the field of product modelling are its flexibility, openness and the richness of interaction possible between many human participants in the modelling environment. Our case studies show the potential for use of our tools and methods, and demonstrate that EM can be an alternative and appropriate approach in the development of tools aimed at product modelling. Moreover, EM is essentially informed by the knowledge about the referent acquired during the model construction. This knowledge is used to extend the model and evolves as the modelling process progresses. The evolution of a user's knowledge can be monitored in an EM environment so as to evaluate what is being learnt about some referent. Thus, since EM artefacts are always open to subsequent extension, refinement and revision, they can also be exploited as interactive environments for learning.

7. ACKNOWLEDGEMENTS:

The authors are most grateful to Richard Cartwright, Chris Roe and other colleagues of the Empirical Modelling Group for constructive contributions they have made.

8. REFERENCES

[1] Alford, R.C. "The IDE Hard Disk Drive Interface". Byte, 317-323, March 1991.

[2] Allderidge, J., Beynon W.M., Cartwright, R.I., Yung, Y.P. "Enabling Technologies for Empirical Modelling in Graphics". Proc. Eurographics UK, 16th Annual Conference, 199-213, 1998.

[3] Beynon, W.M. "Empirical Modelling and the Foundations of Artificial Intelligence". Computation for Metaphors, Analogy and Agents. Lecture Notes in Artificial Intelligence 1562, Springer, 322-364, 1999.

[4] Beynon, W.M. "Empirical Modelling for Educational Technology". Proc. 2nd International Conference on Cognitive Technology, CT'97, IEEE Computer Society, 54-68, 1997.

[5] Beynon, W.M., Bridge, I., Yung, Y.P. "Agent-Oriented Modelling for a Vehicle Cruise Control System". Proc. ASME Conf. ESDA'92, Istanbul, 159-165, 1992.

[6] Beynon, W.M., Rungrattanaubol, J., Sun, P-H, Wright, A. "Exploratory Models for Open-Ended Human-Computer Interaction". Research Report RR346, Dept. of Computer Science, University of Warwick, UK, 1998.

[7] Beynon, W.M., Yung, Y.P. "Agent-Oriented Modelling for Discrete-Event Systems". Proc. ICC Coll. Discrete-Event Dynamic Systems, Digest 1992/138, June 1992.

[8] Erens, F.J., McKay, A., Bloor, S. "Product Modelling Using Multiple Levels of Abstraction: Instances as Types". Computers in Industry, Vol.24, No.1, 17-28, 1994.

[9] Euwe, M.J., Schuwer, R.V. "Configuration of Complex Products". Computers in Industry, Vol.21, 1-10, 1993.

[10] Fischer, C.N. "Performance Evaluation of Chained Declustering Strategies for Disk Array Systems". Proc. UKSIM'99: 4th UK National Simulation Conference, St Catherine's College, Cambridge, UK, 14-18, April 1999.

[11] Ganger, G.R. et al. "Disk Arrays: High-Performance, High-Reliability Storage Subsystems". IEEE Computer, Vol.27, No.3, 30-36, March 1994.

[12] Harel, D. "On Visual Formalisms". Communication of ACM, Vol.31, No.5, May, 514-530, 1988.

[13] http://www.dcs.warwick.ac.uk/modelling.

[14] http://www.hyperfun.org

[15]http://www.lar.ee.upatras.gr/icims/noewww/sotastc.htm

[16] Ness, P., Beynon, W.M., Yung, Y.P. "Applying Agent-Oriented Design to a Sail Boat Simulation". Proc. ASME Conf. ESDA'94, London, 1994.

[17] Roe, C., Beynon, W.M., Fischer, C.N. "Empirical Modelling For The Conceptual Design And Use Of Engineering Products". In these proceedings.

[18] Sun, P.H., Beynon, W.M. "Computer-Mediated Communication: a Distributed Empirical Modelling Perspective". Proc. of CT'99, San Francisco, August 1999.