KNOWLEDGE MODELLING

Suwanna Rasmequan
Burapha University, Thailand
suwanna@compsci.buu.ac.th

Steve Russ
University of Warwick, United Kingdom
sbr@dcs.warwick.ac.uk

Abstract

After some assessment of the problems with current modelling methods the paper introduces Empirical Modelling. This is a new approach to the use of computers for modelling that takes experience as fundamental and advocates the development of methods and tools that seek to compare experience of the real world domain with experience of computer-based models. This contrasts with conventional approaches where experience of an application is represented by abstractions suited for computer optimisation. The rich, interactive kinds of models made with Empirical Modelling methods justify the new term of 'knowledge modelling' in place of knowledge representation.

Keywords: Knowledge, Modelling, Experience, Dependency, Agency, Observables

1.0 Problems of Computer Support

We shall motivate a new approach to modelling and to knowledge representation by summarising some problems of current computer use in the area of business support systems. Since the first development of computer-based support systems and their use as administrative tools by people in business in the 1950s and 60s, there has been an enormous change in their speed and power due to the dramatic changes in technology. But there has not been a corresponding development in principles for software or tools. Some of the practical difficulties of current systems are as follows.

1. There is a sharp separation between users' experience of the world and their experience of computer applications. This is because of the strict structure required by program conventions such as the use of data types, data formats and control statements. That is, to maximize the computing capacity of the machine, instructions, inputs and outputs are in the form that best fit the processes of digital representation. These do not fit well with the understanding of human beings or with the way they experience the world.

2. The passive and limited interaction possible restricts the opportunity to develop new knowledge. Efficiency in collecting and manipulating data is at the cost of allowing new ways of interpreting and interacting with the data which in humans is often the way new knowledge is gained. A system designer may not be aware of the significance of choices of interaction that can affect a particular user's cognitive process. For example, in interacting with other people, or the world in general, the openness and flexibility of individuals' interpretation and interaction is often vital to their learning. Thus a pilot trainee can use a computer simulation to gain significant new knowledge of how to fly an aircraft. But such new knowledge is bounded by the context set by such a system. The context is set by the range and number of variables used, the nature of the interface and controls, and the detail with which the environment is modelled. That is why after a period on a flight simulator a pilot trainee still needs extensive actual flight training before flying any commercial aircraft. A similar interactive system example is the expert system. For example, a trainee doctor may learn new knowledge from a Disease Diagnosis Expert System. However, what can be learnt is derived from what has been put in and the scope of the interaction allowed. A trainee doctor may gain new knowledge of the possible symptoms for a disease. However, it is difficult to deduce any association between a recently developed symptom which has not yet been diagnosed and put in the system with what has been observed by a doctor. In spite of the fact that such systems enable interactive experience, such experience lacks the possibility of open-ended experimenting. Here 'open-ended' means going beyond the pre-conceived context in the above sense, developing new contexts, and acting opportunistically. This is because, in such a kind of system, the interpretation of any interaction is predetermined. Open-ended experiment, which is rooted in open-ended interaction, is a unique and important source of learning. The difficulty of open-ended experimenting is part of the nature of conventional computer-based support systems. This is owing to the emphasis on automatic operation in existing computing paradigms.

3. Support systems are monolithic and neglect the need for adaptability to cultural or individual characteristics. Social factors are undoubtedly a great influence on business practice, which is a social system. For example, norms, attitude, culture, tradition, beliefs, perceptions, practice and values, are always different in different social systems. These factors shape the understanding or interpretation of humans towards particular thing(s) or situation(s) [1]. An example of the significance of difference in cultures is as follows. Consumer rights are not so well protected in Thailand as compared to England, thus a marketing strategy which works in England might not be effective in the Thai market. This kind of thing, if it is not well understood, could lead to constructing either a poor strategy plan for the company or a poor support system. This sort of fact has not been taken into account in most of the current approaches that are used to develop computer-based support systems for business activities. This is especially relevant to those countries that are not leaders in computing developments. For example, most of the leading business firms in Thailand have to employ computerised support systems that were originally developed for their headquarters abroad. Staff are concerned that they must familiarise themselves with the structure of systems that were not designed for them. In general, most of the manual systems with their individual adaptations which firms employed, lost their individual character when they were transformed into computerised systems and they then all had the same structures and functions. The constraints imposed by conventional ways of computing do not allow for the preservation of individual peculiarities. To maximise the full capacity of the machine at the expense of losing responsiveness to the individual is no longer necessary.

Such problems as these are just symptoms of a profound mis-match between what is needed by business support systems and what is available in current computing paradigms. In business this mis-match has led to disillusioned managers and frustrated IT workers. The promises and investment in the 'IT revolution' of the early 1990s have not delivered what was expected. We suggest much of this problem is due to confusion between knowledge (as something humans may have) and information (something computers may process). The term 'knowledge representation', widespread in AI for many decades, has probably done much to foster this confusion. Coupled with the widespread association of computing with automation this has led to an expectation of 'automated solutions' to modelling problems. Instead we shall propose 'integrated solutions' - solutions achieved through the integration of human processes with computer processes and here knowledge plays a crucial role. We now consider the role of knowledge in business needs and computer systems.

1.1 Business View of Knowledge

There are many ways of approaching some classification or comprehension of knowledge. One broad and widely used distinction is between explicit or articulate knowledge and implicit or inarticulate knowledge. Explicit knowledge is transmittable in formal and systematic language and is dealt with in knowledge-based systems. Implicit knowledge or tacit knowledge is deeply rooted in action, commitment and involvement in a specific context [2]. It has a personalised quality, it involves cognitive and technical elements and also involves context. Baets points out that, in the management context, tacit knowledge is the kind of knowledge that deserves most attention. This is because, " *... tacit knowledge gets used in managerial tasks and tacit knowledge is the knowledge which makes the difference. Key to acquiring tacit knowledge is experience. ... It proves extremely difficult to extract this kind of "knowledge" ... The capacity of an organisation to take effective action is based on tacit corporate knowledge ..."* [3, p. 54]. Thus tacit knowledge is acquired through experience and Baets suggests that to deal with tacit knowledge, a number of cognitive elements are involved.

He refers to 'mental models' as defined by [4] as the central cognitive element for managerial problems. These are working models of the world, formed by human beings, by creating and manipulating analogies in their mind. Baets (summarising Senge) refers to the description of mental models, *"... as deeply held internal images of how the world works which have a powerful influence on what we do because they also affect what we see ..."* [5]. Further he cites Kim's description in 1993 that, *"Mental models provide the context in which to view and interpret new material and they determine how stored information is relevant to a given situation ..."*. He himself suggests that, *"... mental models represent a person's view of the world, including explicit and implicit understanding ..."*. He argues that it is not the reality that matters, but the perception of that reality. He links the perception of reality to the significance of the context of learning and knowledge. He argues that individual intelligence can hardly develop knowledge that goes beyond the capacity of the individual, in a system of individual elements. Such knowledge is not stored, it is created each time that there is an interaction of the different elements [3, p. 33].

Thus according to Baets business activities need support for the conbination of explicit and implicit knowledge that is typical of mental models. Systems are needed that can support the growing of knowledge instead of merely the storing of knowledge.

1.2 Computational View of Knowledge

Computer scientists try to represent different kinds of knowledge, e.g. knowledge in an accounting process, in disease diagnosis and in aviation, by different kinds of representational systems that are supposed to suit well the distinct needs of each domain.

Knowledge representation has been extensively discussed and debated in the area of Artificial Intelligence. Since *"... knowledge seems to be a prerequisite for any kind of intelligent activity ..."* [6], people in AI are focusing their researches on ways of representing different kinds of knowledge in computer systems. For example, how to represent strategic knowledge in a chess program; how to capture an expert's knowledge in condition-action pairs; how to make sense of images on the basis of brightness, texture, edges and surfaces etc; and how to model natural language understanding on the basis of syntax, word meanings and their connotations. Such domain knowledge is, in each case, huge and complex. If it could be completely derived and stored in any computer system, it would be very expensive and might still not be usable effectively. There are further difficulties for AI in representing knowledge. Such as how to store the enormous amount of common-sense background knowledge which a person may have [6] and how to represent or incorporate inarticulate knowledge such as tacit knowledge, skills knowledge or experiential knowledge.

The classic application of computing to knowledge representation is in the construction of expert systems. Some of the well-known problems are:

1. Eliciting the knowledge of an expert may be hard or impossible, in which case it cannot be stored in a conventional system of production rules.
2. People can adapt what they already know to completely new situations. In this case new knowledge is gained through open-ended interaction with the environment. That is, people are able to develop new knowledge through experience, this is knowledge that could not be deduced from existing knowledge by any logical means.
3. People's judgement is faster than computer operations. This is owing to their ability to discriminate efficiently among choices. Such discrimination includes cases where human concern is a priority, where relevance has to be decided, where more realistic choices are needed and where economy of scale operates. It is difficult to give computer systems any degree of discrimination; typically they either search exhaustively, or as directed by some pre-defined criteria.

We propose here an alternative approach to the use of computers for modelling that has been applied to many domains such as environments for games design, engineering design, and education and business applications. This approach is called Empirical Modelling (EM). A detailed comparison of this approach with traditional AI approaches and expert systems is given in [7]. For the support of creative and design activities preserving the personal element in modelling is important. For instance, modelling may be guided by assumptions made by an individual that depend on their own experience of a domain and their own vision. Since business activities depend particularly on human and social factors the development of useful business models needs methods that support such factors. Some detailed consideration and illustration of how EM can offer this support in the areas of decision support systems and business process re-engineering are given in two recent theses [8] and [9].

2.0 Empirical Modelling

Empirical Modelling (EM) is the theme of a research group at the University of Warwick, United Kingdom, that has been developed under the leadership of Dr Meurig Beynon since the early 1980s [10]. EM is a broad approach to computing that is both principled and practical. The approach is human-centred in its emphasis on experience. Modelling using EM principles means constructing artefacts on a computer that afford experiences comparable to the modeller's experiences of the world. Typically such experience is generated, interpreted and tested through interaction, which therefore has a central place in EM. The main focus of the methods and tools of EM is with activities that precede (or are assumed as unproblematic in) the standard phases of programming. Such activities include:

1. domain analysis - for example, establishing the relevant entities and mechanisms in a domain and the meaning and significance of terms and processes, and learning how these are affected by context and viewpoint;
2. understanding and identification of problems and tasks; and
3. establishing relevant stakeholders in an enterprise and their initial viewpoints and requirements.

These activities are essential preliminaries to developing software solutions. For small tasks in well-understood areas they can, perhaps, be taken for granted. For large-scale tasks, however, or for tasks that are not yet well understood, they are difficult and have long-term consequences. In the conventional practice of programming, these activities are conducted, for example, by means of extensive listening and discussion, documentation, diagramming, natural language and specialist notations. While performing such activities, the thinking processes proceed mainly by imagination and mental modelling. Some of these methods may also be computer supported. What does not typically happen at this stage is to make rich, computer-based, interactive representations of a domain or system, where these representations can be directly experienced. Building such representations that offer direct experience, encourage active engagement and spontaneous involvement, is exactly the aim of the approach and tools of EM. A large number of such models have been built - mostly by students - which demonstrates that this aim can be achieved even with fairly rudimentary tools.

The way in which a computer-based system is built up in the EM approach depends on construing a phenomenon (i.e. forming a personal analysis, or understanding, of it) in terms of three basic kinds of entity: observables, dependency and agency in the following way. First of all, the entities within the system identified by the modeller as meaningful and relevant are observables. For example, a 'room' may itself be an observable, so may be the 'colour' of the room and whether it is 'tidy', or 'has character'. Sometimes the modeller can discern, or guess at, an indivisible dependency between observables such as that expressed by the rules of a game or by physical constraints or laws. Such dependencies can be expressed by definitions and are then automatically maintained rather like the formulae of a spreadsheet. Then sources of change in the values of observables are identified as agents. These may be parts of the environment (gravity, or just a wall that a ball may bounce off), devices such as an engine or a thermostat, or human participants in the model. There will generally be a group of observables associated with an agent. The agent is able to initiate the state change of its elements and to effect a state change in the phenomenon as a whole.

The structures built up of observables, dependencies and agents are based upon the modeller's viewpoint and interpretation of the domain or, in other words, the modeller's experience of the domain. The general principle of development for a model or artefact in EM is the correlation of one experience (of the domain) with another experience (of the artefact). Whenever an observation or behaviour in the model does not correspond with the domain then new observations or redefinitions may be introduced into the model. In order to establish the reliability of a component of the model experiments are conducted on the component until there is sufficient coherence between the behaviour of the component and the behaviour of its real world referent. Because the basic concepts used in building EM artefacts - observables, dependency, agency - are quite fundamental for the way people make sense of the world they are much closer to human cognitive processes than the abstract data types and control structures familiar in conventional programming. It is therefore natural to regard models constructed using these principles as 'cognitive artefacts'. This is a term introduced in [11] and we use it here in similar way to Norman although we usually have in mind computer-based models. In contrast a mathematical or 'program' model has an abstracted quality in which the possible interactions and modifications are preconceived and circumscribed. While this allows accurate prediction of the model behaviour, it means the model is detached from its context and is hard to revise or re-interpret. From the point of view of new experience or growth in knowledge such a model is 'frozen' and no longer in living, active connection with its referent.

A computer-based system viewed as a cognitive artefact offers a distinctive quality of interaction compared with one viewed as a virtual machine composed of abstract objects. The concept of cognitive artefact refers to a specific instance which may be represented pictorially in an EM model. The concept of abstract object expressed in mathematical and logical statements in a conventional program does not refer to a specific instance. So cognitive artefacts in an EM environment offer an experience-based kind of interaction. Experience of the cognitive artefact is representative of experience of the real world. For example, a pictorial representation of the model offers a kind of direct association of having interaction with each component or the whole model, in the way a car designer interacts with parts, or the whole body of her car model. That is, she can interact with her car model as if she were having actual interaction or experience with it. She can represent her experience in the real world with another experience with the car model.

Cognitive artefacts provide the user with a representation of the current status of their referent. This means, in the case of a problem solving process, that such a cognitivve artefact could support the process through unlimited interaction and may lead to knowledge of new facts about the situation. This is based on the fact that, to most people, a cognitive artefact, which is often represented by a visualisation in EM for the system or situation being studied, enables physical contact (e.g. seeing or touching) and thereby complements the logical power inherent in a more abstract model. An artefact model as a representation of a system or situation is a more accessible medium of human experience than a conventional program.

2.1 Knowledge Modelling in EM

Some of the problems of conventional knowledge representation have been mentioned in 1.2. The very idea that knowledge can be represented in some inert fashion on a machine is contrary to the use of 'knowledge' in everyday language, where knowledge requires a human knower. We have argued in 2.0 that the EM approach to modelling gives rise to models that merit the term 'cognitive artefact'. Because such models are regarded as continuously evolving products of human and computer-based processing they have a stronger claim to be viewed as true knowledge representation. But in order to distinguish them from the AI usage and to emphasise their rich interactive qualities we propose here the term 'knowledge modelling' for the creation and development of such artefacts in EM. Knowledge modelling is based on the close integration of human cognitive processes and computing processes: this is a major difference between EM modelling and conventional software systems or expert systems development.

The significance of the EM environment is that human sense-making activities interact closely with the data manipulation activity. Therefore the significance and flexibility of human interpretation, which is mainly based on common sense or a priori knowledge and experience, are not lost, in contrast with systems that allow the manipulation of symbols only by rules stored in the machine. In reading books, human subjects are included in the processes of manipulating the data, but the lack of facility to exhibit their thought processes externally prohibits them recording (and possibly even realising) much of what they are doing in their thought processes or their knowing. Familiar tools such as paper and pen are artefacts to assist people in their thought process or cognition. They support the cognitive process by allowing users to illustrate visually and freely what they know, in the way they understand it. That is to say, people need cognitive aids to support and extend their cognition.

There is a qualitative difference in the medium for representation provided by EM from the usual means available such as mathematics, languages of various kinds, diagrams etc. As with anything with a certain amount of novelty this difference is more easily experienced than described. By the 'medium of representation' here is meant the combination of an interpreter supporting concepts of agency, dependency and observables together with some means for visualisation such as a windowing display system. The richness and openness of the medium shows itself in the unbounded way that layers of detail can be added to a model at any stage in its development or use. The medium is 'interactive' in the very strong sense of allowing experiments and engagement, in principle at least, with the same freedom that applies to the real-world referent. And the sheer speed of response for the user to see how a modification affects the model leads to an unusual sense of immediacy and directness of experience. In these ways - richness, openness and interactivity - we can compare an EM model with a physical model like a model ship in a wave tank. It is fundamental to the way an EM development proceeds that the modeller's experience of the model is of the same kind as their experience of the referent. This is not the case when the modelling medium is largely symbolic [12].

In the EM environment, human agents are able to build models which are like paper and pen on a grand scale: it offers a new modelling medium allowing, in principle, the immediacy of mental models coupled with similar feedback as from a physical model. This allows the user to control and monitor the manipulation and interpretation of data and so also to exhibit their thought processes and cognition. Thus there is real knowledge representation in an EM environment in which many kinds of knowledge can be integrated in the modelling as they are in human mental modelling.

2.2 Knowledge and Dependency in EM

A key feature of the development of artefacts in EM is the identification and commitment to appropriate dependencies in some domain. In a spreadsheet the dependencies are usually of a simple kind relating numerical values such as quantities of money or marks. EM can be thought of as a radical generalisation of spreadsheets in which the dependencies can be of quite arbitrary kinds and complexity. Nevertheless a great deal of the state change in the real world seems to be reliably accounted for in terms of dependencies. For example, the position of a table lamp may be given by a definition such as

$$\text{lamp\_position} = \text{table\_position} + \text{offset}$$

under the assumptions of typical conditions of gravity, friction, style of lamp etc. Such a dependency models our knowledge of the lamp and table in an interesting way. Not only does it include propositional knowledge (or knowledge by description in Russell's famous distinction) but at the same time it represents a persistent one-way relationship. This relationship knowledge might be experiential knowledge ("I have always observed this."), and so 'knowledge by acquaintance', or the prediction of a theory ("In the presence of gravity, and suitable friction, this is what will happen.") or it might be a matter of design.

Such a definition is clearly different from a mathematical equation such as the typical formulation of Newton's second law that is often expressed in the equation

$$\text{Force} = \text{mass x acceleration}$$

where each quantity has an equal right to be made the subject of the formula. In contrast, a definition is a one-way constraint. For example, in the vehicle cruise control system (see [10]) Newton's law appears in the following form

$$\text{accel} = (\text{tracF - brakF - gradF - rollF - windF}) / \text{mass}$$

with the meaning that as the throttle is opened and the tractive force due to the engine is increased then the acceleration will automatically increase. But not conversely. To calculate the force acting on a pilot due to the plane's acceleration a definition is needed of the form

$$\text{force\_on\_pilot} = \text{mass x acceleration}$$

The choice of observables in a model and the identification of agents each also contribute to the knowledge modelling process of building an EM artefact. The way in which these modelling elements relate to traditional classifications of knowledge needs further work; for example, it would be fruitful to understand better the connections of ’dependency knowledge’ with the declarative knowledge typical in program specifications.

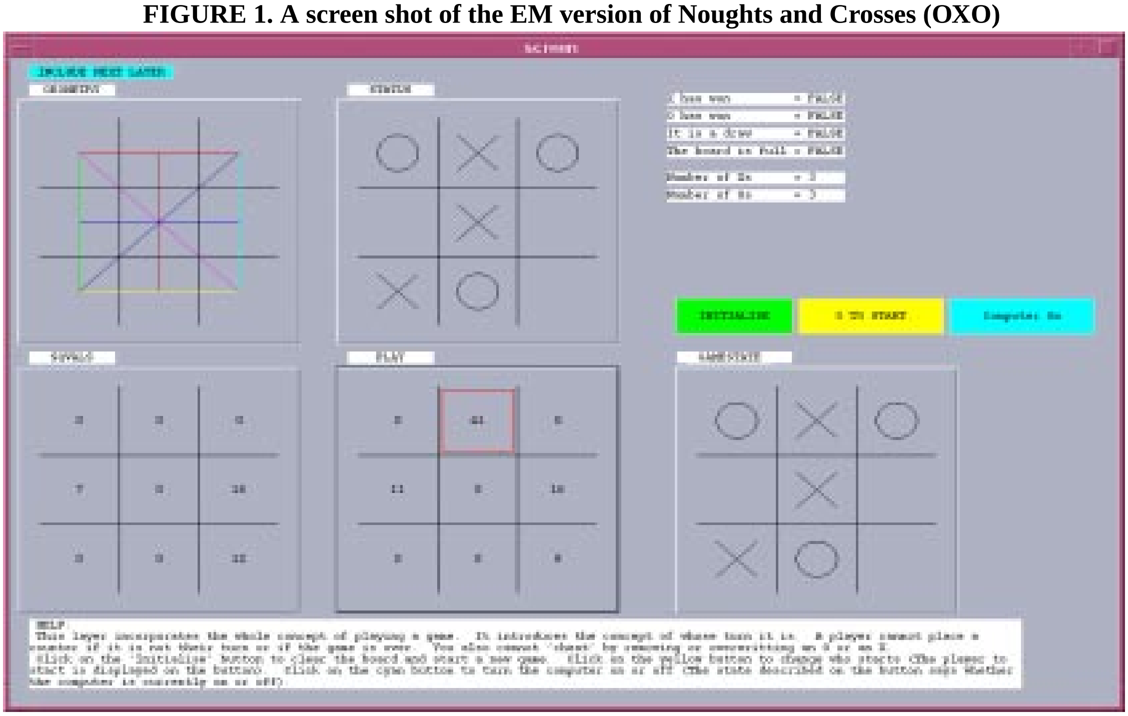
2.3 A Kind of Interaction that Evokes Knowledge

In the EM environment, there is not only a chance to utilise one’s experience, but also a chance to explore new experiences which is often a source of enhancing one’s knowledge. This is possible because of the two-way interaction allowed in the EM environment. This two-way interaction can be compared with interactions like holding a conversation, driving a car or performing artistic works. The essence of this kind of interaction is that the open, immediate feedback in response to individual action provokes a rapid and direct response in tune with that action. The two-way interaction is not the same as the interaction between a modeller or user and a conventional computer model, in which the interaction is preconceived and cannot lead to accruing new knowledge.

In contrast, reading is an example of one-way interaction - that is, there is no response from the other party in the interaction. Nevertheless, this kind of one-way interaction can lead to the use of one’s knowledge and the possibility of gaining new knowledge. This is because, when people are involved in imaginative, creative and analytical thinking, they consult their experience to construe or interpret the things they are encountering or interacting with. However, the lack of an uncircumscribed response from the other party reduces the chance of one’s knowledge being evoked deliberately.

The contrast between one-way interaction and a two-way interaction environment is illustrated in an EM model that has been developed in connection with the implementation of Noughts and Crosses (OXO) (see [13, 10]). With reference to Figure 1, "A screenshot of the EM version of OXO, the first window (‘Geometry’) introduces the concept of lines on the grid. Each winning line is represented by a different coloured line. These are animated to more clearly depict the winning lines. The ‘Status’ window introduces the concept of pieces. Xs and Os can be placed on the board at random. There are no rules at this point. Counters can be overwritten or erased. Several identical counters can be placed in a row (i.e. you do not have to take turns in placing a counter). The ‘Sqvals’ window starts to introduce values for each move into the model. Each position on the board displays a value. This is a score for moving on that square. These scores are calculated by considering each possible winning line which goes through that square. When considering a possible winning line, it looks at how many counters of each type are on the line already. It then gives that line a score. The window labelled ’Play’ shows the scores the computer evaluated on its turn. The square it chooses is highlighted by a red square. The window labelled ’Gamestate’ is the window you actually play in and it introduces the concept of having turns and abiding by the rules of play.

FIGURE 1. A screen shot of the EM version of Noughts and Crosses (OXO)



A conventional version of OXO (say in Java) is an example of one-way interaction, while the EM version represents two-way interaction. They both represent knowledge of how to play the game. However, these two systems represent distinct kinds of knowledge. The significant difference in the quality of knowledge is based on the kind of interaction. That is, free and uncircumscribed interaction that allows users to experiment with the system like scientists can do in their laboratory experimentation, is provided in the EM environment. Through open-ended interaction, a person is allowed to freely experiment with a subject of interest. This very act of unbounded investigation raises opportunity for one to make incidental or radical discovery that adds to one's knowledge, that is to say, one gains new knowledge. The Java-based version in which the interaction with the system is circumscribed, can provide only the intended knowledge of how to play a game, as a book may provide to its readers.

The inclusion of experiential knowledge in the modelling process makes the process of EM significantly different from conventional modelling and programming.

3.0 Conclusion

This paper has proposed that an EM environment offers a new approach to the process of modelling. As discussed in the previous section, an EM environment encourages a special kind of knowledge representation that merits being called knowledge modelling. This includes the combination of propositional, tacit and experiential knowledge, so that the modelling method of EM can support both the products and the processes of knowledge creation. EM models are used as 'tools for thinking' in Pidd's terms [14]. Modelling using EM is similar to the way designers develop physical models to represent their mental models of the domain under study. EM models offer a good medium for modellers to interact with artefacts and make qualitative judgements about them in place of the real world domain. The authors are pleased to acknowledge the incorporation in this paper of insights arising from discussions with Meurig Beynon and many other members of the Empirical Modelling group at Warwick and reference to models built within that group.

4.0 References

[1] E. Goffman. Role distance (originally published in Encounters: Two studies in the sociology of interaction, 1961). In C. Lemert and A. Branaman (eds), The self and social roles, *The Goffman reader*. Blackwell, 1997 (reprinted 2001), pp. 40-42.

[2] I. Nonaka. A dynamic theory of organizational knowledge creation. *Organization Science*. Vol. 5 No. 1, 1994.

[3] W. R. J. Baets. *Organizational learning and knowledge technologies in a dynamic environment*. Kluwer, 1998.

[4] P. Johnson-Laird. *The computer and the mind.* 2nd Edition. Fontana Press, 1993 (first published in 1983).

[5] P. M. Senge. *The Fifth Discipline: The Art and Practice of the Learning Organization*. Doubleday, 1990.

[6] E. C. Way. *Knowledge representation and metaphor.* Intellect Books, 1994.

[7] W. M. Beynon. Empirical Modelling and the foundation of artificial intelligence. In C. Nehaniv (ed.), *Proc. of International Workshop on Computation, Metaphor, Analogy, and Agents*, University of Aizu, Japan, April 1998.

[8] S. Rasmequan. *An approach to computer-based knowledge representation for the business environment using Empirical Modelling*. Ph.D. Thesis, Department of Computer Science, University of Warwick, UK, 2001.

[9] Y. C. Chen. *Empirical Modelling for participative business process reengineering*. Ph.D. Thesis, Department of Computer Science, University of Warwick, UK, 2002.

[10] Empirical Modelling Web Site - http://www.dcs.warwick.ac.uk /modelling

[11] D. A. Norman. Cognitive Artifacts. In J. M. Carroll (ed) *Designing interaction: psychology at the human-computer interface*, Cambridge University Press, 1991.

[12] S. B. Russ. Empirical Modelling: the computer as a modelling medium. *Computer Bulletin*, April 1997.

[13] W. M. Beynon and M. S. Joy. Computer Programming for Noughts-and-Crosses: New Frontiers. In *Proc. PPIG'94*. 1994.

[14] M. Pidd. *Tools for thinking: modelling in management scienc*e. Wiley, 1996.