



The seven habits of highly defective simulation projects

JD Salt

General Dynamics UK Limited, Bryn Brithdir, Oakdale Business Park, Oakdale, Blackwood, South Wales, UK

Seven classes of persistent mistake are identified and described, based on the author's 20 years' experience of making mistakes in simulation modelling. The errors identified are trifle-worship, belief in answers, connectionism, the black box mistake, methodolatry, the dead fish fallacy, and the Jehovah problem. Each is discussed in turn. It is expected that fellow practitioners will recognize many of these mistakes, and hoped that this paper will help to avoid repeating them. *Journal of Simulation* (2008) 2, 155–161. doi:10.1057/jos.2008.7

Keywords: simulation; practice of OR; methodology

1. Introduction

Eighteen years and 15 million copies ago, Stephen Covey wrote 'The seven habits of highly effective people' (<http://www.whitedovebooks.co.uk/7-habits/7-habits.htm>, accessed 1 September 2007), establishing himself as one of the most successful business and personal-improvement gurus of recent times. These seven habits are:

- Habit 1: Be proactive (in the sense of Victor Frankl's original coining, rather than a trendy synonym for 'active').
- Habit 2: Begin with the end in mind.
- Habit 3: Put first things first.
- Habit 4: Think win-win.
- Habit 5: Seek first to understand, then to be understood.
- Habit 6: Synergize.
- Habit 7: Sharpen the saw (personal development and reflection on experience).

He has recently (2004) added an eighth habit, with bonus DVD. These habits may be regarded as recommendations for adoption by those eager for personal success. My aims in this paper are very much less ambitious. I offer no recipe for personal success in life, nor advice on how to develop your character. The scope of my remarks is limited to the field of stochastic discrete-event simulation, and I offer only hints at how to avoid failure, which is not quite the same thing as a recipe for success. The astute reader will have noticed that the weak joke in the title is merely an excuse to offer, as others have before, a list of my favourite simulation modelling mistakes. Averill Law has listed what he calls 11 critical pitfalls in simulation modelling (<http://www.averill-law.com/simulation-news-papers.htm>, accessed September 2007).

Correspondence: JD Salt, 43, Brookland Road, Risca, Caerffili, South Wales NP11 6BH, UK.
E-mail: jdsalt@gotadsl.co.uk

Ed Russell listed 10 reasons for simulation project failure, and his list has perhaps been the most influential (McLeod, 1982; Russell, 1983). Slightly different in flavour are the six rules offered by Gene Woolsey (2003), which are mainly advice on things not to do, and applicable to OR generally rather than just to simulation modelling. Ray Paul has presented what he calls the 'diversionary triathlon' (Taylor *et al.*, 2004) of concerns that distract simulation modellers from their proper business and so inhibit success.

What are presented here are seven habits of mind that I have observed from experience, that are likely to have baleful effects on a simulation project. I have not included common mistakes of management and life in general, such as failure to select and maintain an achievable aim, failure to adapt to changed circumstances, or the pervasive 'failure of communication' on which legion other failings may conveniently be blamed. Some of the points chosen are perhaps of more widespread applicability to operational research or information systems in general; others are quite specific to simulation modelling. For each of the habits chosen, I will describe the belief it entails; say why this is a mistake; account for possible motivations for the habit; and describe its symptoms and effects.

2. Trifle-worship

In a previous invited talk (Salt, 1993), I coined the word 'bagatellomania', and several others still uglier, to describe this phenomenon. It is the habit of unrelenting pursuit of detail, and of proliferating entities beyond necessity; the apparent worship of mere trifles.

2.1. The belief

The belief of trifle-worship is 'the more detailed the model, the better'. Not only is this belief widespread, it attains at

times the blessing of official sanction, as for example the DMSO special topic paper on fidelity (<http://sysdyn.clexchange.org/sdep/Roadmaps/RM9/D-4101-1.pdf>, accessed 1 September 2007), which describes fidelity as a 'primary measure of goodness' for a simulation.

2.2. The mistake

Trifle-worshippers confuse detail with realism. Although one very frequently hears people talk of 'realistic models', it should be recalled that 'realism' is a term that comes from the world of art and theatre, not from mathematics. The graphs showing realism increasing linearly with detail cannot have their axes marked, for there are no SI units of realism. The function of simulation models (at least, those used for decision support, rather than for training) should not be to get users to suspend their disbelief, but rather to assist them in understanding the system under study. In extreme cases, users may suspend their disbelief so far as to do away with it altogether, and end up mistaking the model for reality (see the Jehovah problem, below). The inclusion of 'mere corroborative detail, adding verisimilitude to an otherwise bald and unconvincing narrative' often serves only to obscure the essentials. Ed Russell (1983) identified 'inappropriate level of detail' as one of his top 10 simulation errors, and 'inappropriate' here almost invariably means 'too much'. It is usually easier to add detail to a model than it is to subtract it out once added, so erring on the side of too little detail is easier to correct than too much.

2.3. The motivation

One motivation for trifle-worship is to blind people with science and conduct proof by intimidation. Related to this is the widespread bad habit of software developers to show off the fruits of their labours by pointing to a long list of 'features', hoping that suggestible people will mentally elide these with 'benefits'.

More tolerably, one might include excessive detail in order to reassure oneself one is leaving nothing out. This is a fool's errand, for simplification is the essence of modelling, and if it were possible to create a full-fidelity model of a system it would presumably be no easier to study than the original. It is inevitable that a system being studied by simulation will be imperfectly understood, and it is reasonable for a simulation model to be used to explore the question of which factors are important and which not. Under these circumstances it is probably unavoidable that one will spend time creating aspects of a model that turn out not to have a big influence on the final result. That, however, seems not to fall under this heading, as at least a brief excursion into that detail was necessary if only to establish that it was not decisively important. A conscientious modeller, having designed for 'unpluggability' as well as 'pluggability', would then remove the unnecessary detail.

Finally, office politics may demand the creation of large and expensive models in order to build an empire with the resources demanded by such a scale of effort.

2.4. Symptoms and effects

Symptoms of this mistake include having an ever-growing model, and an unending need to find more data to feed it. Indeed the model may never be finished; it is not unknown for an ambitious simulation project to result in the creation of a lavishly equipped and well-staffed laboratory, and for the person in charge of such an effort to have been promoted amid general acclaim before a single useful simulation model has been completed. If a model is completed, it may become so complicated that it is hard to distinguish the signal from the noise, and considerable digging may be found necessary to determine whether an observed effect is a genuine result of the intended model behaviour, or a 'model artefact'. Modifying the model to reflect changed understanding becomes a horrifying prospect, especially if the detail is not adequately documented.

3. Belief in answers

One of the naïve beliefs often apparently held about simulation modelling is that it can reliably and accurately predict the future. Of course, answers can be predictive, correct, and entirely useless, as for example the prediction that it will rain in South Wales some time next month; but people seem to expect simulation to do better than that.

3.1. The belief

There is an answer. A simple answer. To life, the universe, and everything.

3.2. The mistake

There may be 0 or ∞ answers, and they may change with time.

You might have misunderstood the question you asked, so that, like the oracle of Delphi, your simulation gives you an answer that is misleading. An example of this is a case where a simulation model of air defence was giving answers that were not understood, and so was replaced by a linear programming (LP) model—not necessarily a bad thing in itself, but the LP model was used to optimize the number of attacking aircraft shot down, which indicates a failure on the part of the modellers to understand the purpose of the system under study (the main aim of air defenders is to stop their own people getting bombed).

The point of simulation modelling is, as the strapline of *Simulation* always used to say, 'for understanding'. As Mike Pidd (1993) has pointed out, Ashby's law of requisite variety

seems to require that anything (such as a model) used to make decisions for controlling a system have at least as much variety as the system being controlled. Yet full-fidelity modelling is impossible. The point is that the model alone can never possess the requisite variety; only the combination of model and human brain can have enough variety. It is therefore bootless to imagine that a computer model can deliver cut-and-dried answers that do not require the user to engage his brain. Simulation modelling properly aims to support human decision-making, not to usurp it.

3.3. *The motivation*

The need for answers is understandable given the normal psychological need for closure, and most people would like to have some way of reducing uncertainty about the future. There is also often the practical need to get on with things, and, less appealingly, macho management insistence on 'delivery' and 'results'. People are naturally very bad at reasoning about probability, especially low probabilities (and still more so conditional probability). It is therefore much easier to assume certainty, even where there is none; and in the case of low probabilities, this more easily seems justifiable. For more on our difficulties in reasoning about uncertainty, see particularly Taleb (2001) and Piatelli-Palmarini (1994).

3.4. *Symptoms and effects*

A belief in answers manifests itself in disappointment with models that do not produce clear-cut answers, or raise more questions than they settle. Worse still is disappointment with models that do not produce the particular answer one was expecting; warning bells should ring when a model requester says 'Write me a model to prove...'; models are fine for investigation, but not for advocacy. The fine old Persian tradition of killing the messenger who brings bad news is not yet entirely extinct, but the conscientious modeller should resist any pressure to rig a model to produce the desired answers.

The belief in answers also tends to mean that provisional findings, originally accepted because there was nothing better, gradually gain respectability by mere lapse of time, and eventually become gospel because no further investigation is ever conducted. Probably the most frequent symptom of a belief in answers is a fixation with validation by comparison with historical data. This fixation is so widespread and so entrenched that I cannot adequately dispose of it here, but I refer the reader to a concise and pungent paper by Kleindorfer and Ganeshan (1993).

4. *Connectionism*

The modern fashion is for connectedness, for 'joined-up' thinking. Metcalfe's law says that the value of a telecomms

network increases as the square of the number of subscribers; does the same principle apply to simulation models?

4.1. *The belief*

If one model is good, then having several models connected together is better.

4.2. *The mistake*

Models can seldom be joined up meaningfully. To do so requires that they share substantially the same *Weltanschauung*. Sterman points out that the simplified nature of models requires details to be left out, and 'the purpose of the model acts as the logical knife' (Sterman, 1991). Models that are designed for different purposes are probably missing some of the elements needed to be usefully joined to one another. It may be that the combined system represented by the combination of models can be cleanly decomposed into the sub-systems represented by the individual models, but it seems unlikely. More probable is the experience of Fishwick's warbler. Paul Fishwick tells of a simulation study into the effect of drainage patterns on the breeding success of some kind of reed warbler. A drainage model already existed; and a model of the birds' breeding patterns already existed. When the two were joined together, the result was not a model of how drainage influences warbler breeding, but rather a drainage model with a breeding model stuck on the side.

Distributed simulation is much used for military training, where in contrast to decision-support modelling the idea of connecting training simulators at least makes sense. The world of DIS/HLA is famed for what are known as 'FOM wars', where people representing each of the simulations to be federated thrash out the exact composition of the FOM, or Federation Object Model. I gather that this is a painful process, even where the participants share substantially the same purpose (training); coordinating the disparate *Weltanschauungen* of models written for different purposes would presumably be worse still. Tolk and Muguira have proposed a model of five levels of interoperability (Tolk and Muguira, 2003), the highest being 'conceptual interoperability', or what I have termed sharing the same *Weltanschauung*. It seems to me that unless this level is attained, the other levels are quite useless.

4.3. *The motivation*

Simulation models can absorb substantial amounts of time and money for their creation, and, in the fine tradition of the sunk costs fallacy, people imagine that if you have spent a lot on something, then it must be an investment. This misses the point that, to paraphrase Eisenhower, models are useless, but modelling is vital. If one imagines models to be an asset,

then it makes sense to attempt to re-use them, even though the effort is probably even greater than with ordinary software.

I suspect that the desire to join models together often originates from the ambition, common to beginners in the modelling game, to have a single grand unified model of everything. As Steve Downes-Martin said when HLA was new, 'object-oriented simulation and megalomania are not incompatible'.

Possibly the willingness to expend development effort in joining models up conceals a desire to stay within one's established comfort zone, and concentrate on what one understands (data processing problems) rather than what one does not (simulation modelling problems).

4.4. Symptoms and effects

An obvious symptom of connectionism is the proliferation of people whose main work is in IT 'plumbing' and network administration rather than in essential modelling.

Another obvious symptom is the issue of edicts on adoption of assorted standards, in order to make models interoperable. A particularly topical example of this is the current fashion for misguided attempts to force simulation modelling into the pattern of MODAF, an 'enterprise architecture framework' (<http://www.modaf.com/>, accessed 1 September 2007) after the pattern established by Zachmann (1987).

5. The black box mistake

A 'black box' is part of a system whose inner workings are not visible. This may be a good thing in software development generally, but probably not in simulation modelling.

5.1. The belief

There is no need to worry about the inner workings of a simulation model, just take note of the outputs.

5.2. The mistake

Linked to the mistake of believing in answers, this belief holds that results are more important than the process by which they are obtained. While this is true in some circumstances, it is not in the case of simulation modelling for decision-support; mechanism must be exposed to view in order to help understanding. 'Using mysterious results' is another entry in Ed Russell's top 10.

A related mistake is the belief that modellers must know what they are doing, perhaps because the model has been 'validated', or, worse, 'accredited' or 'certified' (worse still, 'certificated'). Even if it happens that the modellers do know

what they are doing, mechanism should be exposed to ensure that the user shares the same picture of what is going on.

Excessive confidence can be invested in a model because people project things onto it that just aren't there. Just as a few well-chosen lines drawn on paper let us discern the familiar face of Charlie Brown, and even to tell his mood, so people can read far too much into a simple model if the internal mechanism is not visible.

5.3. The motivation

The motivation for the black box mistake from the developer's side may be, in other circumstances, entirely praiseworthy, since concealing mechanism from view is good practice in software engineering. It may, on the other hand, be driven by a guilty feeling that the inner workings of the model are so horrible that they are best not exposed to view.

The black box mistake reposes unjustified trust in delphic pronouncements, and therefore increases business opportunities for snake-oil vendors.

5.4. Symptoms and effects

The most obvious symptom of the black box mistake is seen when model results are presented without any presentation or explanation of the workings that produced them. The results may be in the form of graphs, charts or tables, but it is also possible to conceal the workings of a model behind slick 3-D animated graphics. An example is the use of 'special effects' in synthetic environment 3-D viewers; these are phenomena such as dust and smoke which are not present in the underlying simulation, but are added on solely for display. Anyone viewing such a display would be entitled to believe that the effects of smoke and dust were represented in the underlying model, but this may not be true.

Not being able to see inside the black box may result in mistakenly accepting bad advice, or mistakenly not accepting good advice. Even if the results are not actually refuted, there may be a failure to have confidence in (and therefore act in accordance with) parts of the model that remain behind the curtain, even though they be good. More than once I have seen model results become accepted only once animated graphics were available to dramatize the operation of the system that produced them. This is fine so long as the model is trustworthy and the graphics accurately portray its innards; otherwise, there will be trouble.

6. Methodolatry

The imposition of standard methods and processes to be observed in all aspects of life continues to be a blight on the joy of living, and simulation modelling is no exception. Curiously, some people do not seem to object to this, but

embrace the processes with enthusiasm, and attribute quasi-magical properties to them, whence the word 'methodolatry', the worship of method.

6.1. The belief

If only we follow the method assiduously enough, everything will be all right.

6.2. The mistake

Given the essentially creative nature of decision-support modelling, the obvious mistake here is to imagine that there can be a fixed drill for thinking about new things. Related to this is perhaps the belief that the people who wrote the method are more knowledgeable than you are—which they might be in general, but almost certainly are not about the situation you are in right now.

6.3. The motivation

Just as with connectionism, following an established method may show a laudable desire to profit from others' experience. Having a laid-down method or drill may also give you a way to proceed when you do not know what you are doing, which is often the case in decision-support simulation. Recognizing problem patterns and having a bag of thinking tricks to get you started are both good things, but they become harmful when they turn into unthinking adherence to a fixed process. They are especially harmful when imposed on modellers by an external authority.

Less laudably, one may be motivated by fear of departing from the approved method, or by the need to protect against blame for unpleasant eventualities pungently expressed in the initials CYA (a polite interpretation of which would be 'Cover Your Anatomy'). If you stick to the official process, then even if things turn out badly, you always have the excuse that you followed the method; and some people seem to think that it is better to be wrong in company than right on your own.

Finally, one suspects a conspiracy of clerks to reduce all creative work needing an inquiring intelligence to the level of dull, procedural routine. Such a view is properly mocked in mainstream programming circles as 'creationism', defined as 'The (false) belief that large, innovative software designs can be completely specified in advance and then painlessly magicked out of the void by the normal efforts of a team of normally talented programmers' (Raymond, 1993). Of course, adhering to fixed and predictable procedures should make planning and estimating easier; but it is likely to squeeze out discovery, which rather destroys the point of simulation modelling.

6.4. Symptoms and effects

When methodolatry abounds, a great deal of effort may be expended before any modelling is done in producing plans, 'architectures', 'frameworks' and all sorts of things that fall into the category of 'wampum deliverables' (a name due to de Grace and Hulet Stahl (1990)). While it is true that a certain amount of preparation and planning is necessary, the pernicious attitude of 'doing it right the first time' results in an excess of it. Since the modeller is invariably exploring areas that are poorly understood, no amount of preparation will suffice to avoid making mistakes. Moreover, mistakes made in a simulation are by the nature of things likely to be much less harmful than those made for real. The right course is surely to make the mistakes, learn from them and correct them; mistakes not made at the simulation modelling stage will hurt much more if saved up to be made later.

7. The dead fish fallacy

What makes simulation models interesting and worthwhile is the way they capture the dynamic behaviour of systems. Live systems cannot usefully be studied using dead models.

7.1. The belief

As things have been, they remain.

7.2. The mistake

This fallacy consists in thinking that one can successfully model a dynamic system using a static model. As Peter Senge (1990) says in *The Fifth Discipline*, 'The real leverage in most management situations lies in understanding dynamic complexity, not detail complexity' (emphasis in original).

Stasis can be mistakenly assumed at a number of levels. Presumably, the adoption of simulation modelling implies at least doing better than a wholly static representation, but there are still other layers of stasis. Modellers may concentrate on the steady-state behaviour of a system, when it is transient behaviour that is critically important. They may assume smooth changes where the system is really 'jumpy'. Even with something as well adapted to transience and lumpiness as a Poisson arrivals process, one might assume a stationary arrival rate where it is not justified.

7.3. The motivation

In his *Engineering in the Ancient World* (Landels, 1978), Landels points out that the ancient Greeks and Romans had great difficulty in thinking about *genesis* and *phthora* (*γενεσις*, 'coming to be' and *φθορα*, 'passing away'). I do not think that this difficulty is confined to the ancients. Just as human minds naturally prefer closure and certainty to

uncertainty, so they find it much easier to reason about static systems or representations than dynamic ones.

If reasoning with dynamic representations is hard, creating software for them is harder still, and most of the diagramming methods so far developed in software engineering are of a static rather than an executable nature. In particular, the ubiquitous class diagram (once referred to as the 'extended entity-relationship diagram', showing its familial resemblance to database development methods) is not usually treated as if it were an executable representation.

7.4. Symptoms and effects

The main symptom of the dead fish fallacy is simply missing the point. A fine example of the trap of static thinking about dynamic systems is shown in 'The beer game' in *The Fifth Discipline*. Another striking example is a paper I heard delivered that used a Petri-net-based method to assess the information flow in an air defence system, and hence the system's effectiveness. Petri nets are one of the formalisms capable of adequately capturing dynamic behaviour, and the research question was an interesting one, but unfortunately the author chose to assess the steady-state behaviour of the system. The effectiveness of an air defence system, by its very nature, depends critically on its transient behaviour!

8. The Jehovah problem

Sometimes, people lose sight of the difference between model and reality.

8.1. The belief

I can, as a perfectly accurate external observer, write a single functional decomposition of the entire system under study.

8.2. The mistake

There is no such thing as a perfectly accurate external observer, as explained by Weinberg (1975). It is therefore likely that any system obvious enough for an observer to possess the whole truth about it is not a system interesting enough to be worth modelling. Nonetheless, people persist in believing that such accuracy is obtainable. In extreme cases, modellers forget the difference between their own creation and external reality. If they are physicists, they may even elevate their models to the status of laws.

8.3. The motivation

The source of this mistake might be a reasonable wish to see the big picture, not moderated by a healthy humility about the size of one's own brain.

I suspect that a contributory factor is the subject/object separation that has been an implicit part of most of science since Descartes' time. Personally, I have found that successful modelling more or less requires something like the Zen elimination of subject/object distinctions. Rather than studying the system as a dispassionate and objective observer, one gains insight more quickly by pretending to be part of the system and playing imaginative games to explore it. Such an approach necessarily involves behaviour that may appear undignified, and people may not be comfortable with it.

While it is doubtless an over-generalization, it seems to me that people who try to take a strictly analytic approach to modelling tend to rely on functional decomposition, whereas those who take a synthetic approach prefer an object-oriented decomposition. Because functions do not persist over time, such an approach does not cope well with temporal aspects of a system's behaviour (thus tending also to the dead fish fallacy). However, functions are nicely susceptible to mathematical treatment, in the way that real systems are not.

8.4. Symptoms and effects

Rigour in the application of statistical methods is always welcome in a simulation modelling project. Because simulation experiments are typically exercises in gathering statistics, it is a good habit to quote confidence limits on the results. However, people sometimes forget the meaning of these confidence limits. If I quote the upper and lower 90% confidence bounds for some measure collected from a simulation treatment, I am only saying that I am 90% sure that this measure would lie within these bounds if I could run the simulation for an infinitely protracted interval. I am very definitely not claiming 90% confidence that the same measure in the real world lies between these limits. When people forget this, it is a sign that the Jehovah effect is operating.

Another sign of the effect is when objections are raised to a model because it is 'wrong', as if it were possible for a model ever to be anything else, and the related objection to having two models of the same phenomenon. A common manifestation of this, linked to the connectionism fallacy, is a desire to have a single giant model of everything.

9. Conclusion

Imagine that a project manager starting out on a simulation project thinks as follows:

'If I, as an objective observer of reality, can only apply the correct method to produce a sufficiently detailed and complex model (or set of connected models), then this model will give me definitive answers about real life'.

I do not think that it stretches credulity unduly to imagine a project manager thinking or saying such a thing (I hope a modeller would be more sensible). Yet that short sentence implies, or at least hints at, all seven of the mistaken habits I have discussed in this paper. Look out for people near you saying similar things.

How do we deal with these mistaken habits of mind? There is an old joke about a man who went to the doctor, complaining 'Doctor, doctor, it hurts when I hold my hand up over my head'. The doctor's reply was 'Don't hold your hand there, then'. Merely being aware of the nature of some of these mistaken modes of thought should, I hope, be enough to help sensible people to avoid them.

In some cases, avoidance may not be so easy, as people may have vested interests in perpetuating some of these mistakes, especially where they tend to encourage large project staffs and budgets. I've made all the mistakes listed here except (I think) the Jehovah one, and plenty more besides.

How can success be assured for a simulation project? That is not a question to be answered shortly, if there is an answer at all, although avoiding the seven habits presented here is probably a good start. I will encourage you to be brave, face uncertainty rather than trying to hide it, not be afraid of appearing foolish, get stuck in, and to try to lose your self in the problem. Take for your example Ms Frizzle (http://www.scholastic.com/magicschoolbus/home_2.htm, accessed 1 September 2007), of *The Magic School Bus*: 'Take chances, make mistakes, and get messy'.

References

- De Grace P and Hulet Stahl L (1990). *Wicked Problems, Righteous Solutions*. Prentice-Hall: Englewood Cliffs, NJ.
- Kleindorfer GB and Ganeshan R (1993). The philosophy of science and validation in simulation. In: Evans GW, Mollaghasemi M, Russell EC and Biles WE (eds). *Proceedings of the 25th Winter Simulation Conference*, Los Angeles, CA, ACM Press: New York, pp 50–57.
- Landels JG (1978). *Engineering in the Ancient World*. Chatto & Windus: London.
- McLeod J (1982). *Computer Modeling and Simulation: Principles of Good Practice*. Society for Computer Simulation (SCS): La Jolla, CA.
- Piatelli-Palmarini M (1994). *Inevitable Illusions*. Wiley: Chichester.
- Pidd M (1993). *Tools for Thinking*. Wiley: Chichester.
- Raymond ES (1993). *The New Hacker's Dictionary*. MIT Press: Cambridge, MA.
- Russell EC (1983). *Building Simulation Models with SIMSCRIPT II.5*. CACI: Los Angeles, CA.
- Salt JD (1993). Simulation should be easy and fun! In: Evans GW, Mollaghasemi M, Russell EC and Biles WE (eds). *Proceedings of the 25th Winter Simulation Conference*, Los Angeles, CA, ACM Press: New York, pp 1–5.
- Senge PM (1990). *The Fifth Discipline*. Doubleday: New York, NY.
- Sterman JD (1991). A Skeptic's guide to computer models. In: Barney GO (ed). *Managing the Nation*. Westview Press: Boulder, CO.
- Taleb NN (2001). *Fooled by Randomness*. Thompson Texere: New York, NY.
- Taylor SJE et al (2004). Panel on future challenges in modeling methodology. In: Ingalls RJ, Rosetti MD, Smith JS and Peters BA (eds). *Proceedings of the 2004 Winter Simulation Conference*, Washington DC, ACM Press: New York, pp 327–335.
- Tolk A and Muguira JA (2003). The levels of conceptual interoperability model. 2003 Fall Simulation Interoperability Workshop, Orlando, FL.
- Weinberg GM (1975). *An Introduction to Systems Thinking*. Dorset House: New York, NY.
- Woolsey RED (2003). You too can use operational research! In: Hewitt RL (ed). *Real World Operations Research*. Lionheart Publishing: Marietta, GA, pp 42–47.
- Zachman JA (1987). A framework for information systems architecture. *IBM Systems Journal* 26(3), IBM Publication G321-5298.

Received 28 November 2007;
accepted 8 April 2008 after one revision

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.