TOMORROW, TODAY:
SYSTEM DYNAMICS MODELS OR EXPERT SYSTEMS?

Graham W. Winch
Professor of Business Analysis
Plymouth Business School
Drake Circus
Plymouth
PL4 8AA
England

It was once suggested to me, half jokingly perhaps, that System Dynamics offers "2nd-Generation Expert Systems - before the 1st Generation". This paper reconciles the theories and processes, and draws upon business consulting assignments, to examine how close to reality this notion is.

- What are Expert Systems supposed to do?

A very basic description has been offered by Clifford et al (1986):

> "An expert system is a computer application that provides decision support similar to that of a human expert in solving problems".

or similarly (Hertz, 1988):

> ".... computer programmes that provide advice and diagnoses for advising problems ordinarily dealt with by human experts".

Most expert systems that have been developed to date are in scientific fields, for example, medicine, engineering and geology. Conversely it has been asserted (Coombs & Atty, 1984) that despite significant effort by the Artificial Intelligence community far fewer substantial applications have resulted than might be expected - particularly in the business decision field. Cullen and Bryman (1988) in a survey of 70 expert systems concluded that successful applications are most likely in:

- narrow rather than wide knowledge domains.
- where required expertise is mainly factual rather than procedural.
- systems contain shallow knowledge rather than deep underlying knowledge.

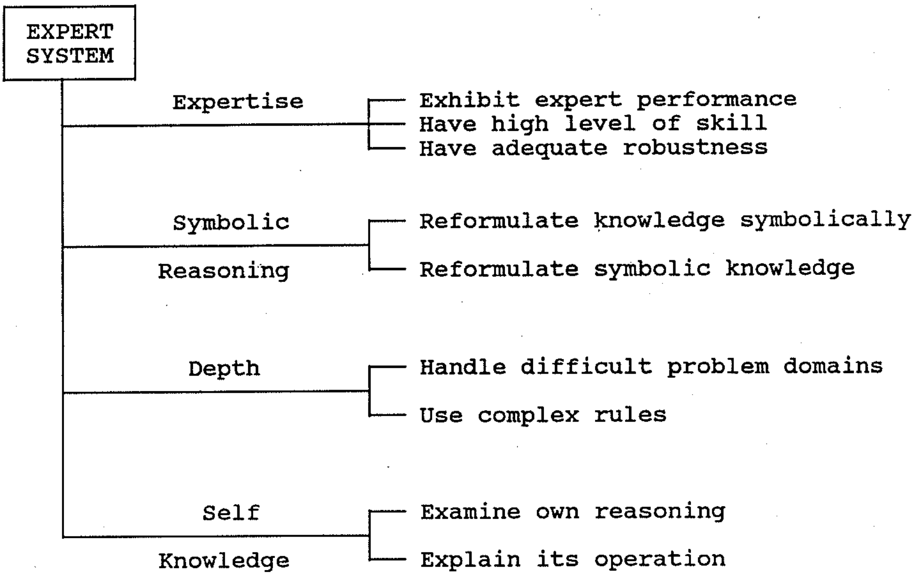
This points to a technology which may have great potential, but with the exception of a few, mainly factually-based scientific systems, has failed to provide many fully fledged working systems which offer true human-like knowledge and reasoning.

- Are system dynamics models expert systems?

In basic terms expert systems are comprised of the following elements:

1. factual and procedural information,
2. rules that describe the relationship between stored data and inputs,
3. methods, programmed within the system, for coming to conclusions based on the stored "knowledge", and
4. an interface with the user for receiving his/her queries, offering results and for explaining the need for inputs and the system's reasoning.

Waterman (1986) has identified key features of expert systems by which he differentiates them from conventional (sic) systems:



On first glance the purpose of a business expert system could be the same as for a system dynamics model, and indeed the processes in model/system development may be very similar (Winch, 1984). However on strict definitions a typical system dynamics model could not be classed as an expert system, as it will inevitably lack a number of these features. That said, many so-called expert systems also lack a number of these features, and have been criticised for brittleness (Hudlicka, 1988), failure to exhibit true expert performance - "overblown accounts of what they can do" (Towris, 1986), and (as discussed earlier) a low presence in difficult problem domains with complex rules.

However, system dynamics models, particularly the large scale business analysis and forecasting models with which the author is familiar, do encompass many of these features. They manifestly contain information concerning facts, identities and procedures within the modelled systems, and rules or inter-relationships that link and inter-link that information with inputs. Indeed recent ES literature has begun to advocate and utilise causal relationships as opposed to "rules-of-thumb gleaned from experience", with the assertion that causal models, because they represent generalised understanding rather than specific problem-solving rules, are likely to be more versatile (Basden, 1984).

Hudlicka (1988), using arguments familiar to system dynamics practitioners, asserts that in constructing causal models for expert systems, ES practitioners are forced to understand the problem domain well enough to express formally its structure and behaviour. He continues that this type of understanding is qualitatively different from that required when knowledge is represented by uncorrelated rules, "as was the case with the first generation of expert systems".

This freedom has enabled the system dynamics modeller to adapt and use generic structures within different business models; for example, plant start-up and shutdown decisions, capital investment decision-making, product and technology upgrading. Apart from the obvious value of "one model many uses" economics, this bring two further benefits:

- these generic structures are continually being verified and modified, representing a learning process.

- a developing model typically combines such existing knowledge with different domain expertise for the new application.

This second point is particularly important in terms of the "self knowledge" feature in Watermans' earlier definition. Although system dynamics models do not generally permit interrogation by the user of the system for explanation of reasoning or input requirements, the comparatively transparent code in DYNAMO, STELLA and other similar syntax languages means that the nature and requirement for particular information can be readily reviewed by the user in both existing generic structures as well as in the developing new domain areas. This self-knowledge aspect is enhanced by causal-loop or influence diagrams, and obviously particularly so by the on-screen flow diagrams with STELLA which vividly portray the knowledge of inter-relationships captured in the model.

DYNAMO, along with other packages with similar syntax, is a non-procedural language. It therefore shares the model building and implementation advantages over procedural languages with declarative languages like PROLOG, which Hampshire (1988) contends are ideal for simulating real-world processes. (Though, even with a declarative language like PROLOG, some find that adding new rules as new knowledge is acquired may need restructuring of the programme (e.g., Clifford et al., 1986) - a rare requirement with DYNAMO). In fact, Hampshire maintains that an expert system is really just another form of declarative language, with only the particular distinction of being able to explain why it comes to a conclusion. It has to be questionable if with current hardware and software any internal reasoning and explanation facility could be built into an expert system that could (realistically) explain loop-gain in typical system dynamics models, or the complex inter-relationships and frequently observed counter-intuitive behaviour of such complex systems. The combination of causal-loop diagrams, transparent code, and the system dynamics modeller does at least provide, currently, a satisfactory hybrid system for achieving this.

In one final, interesting respect system dynamics models may be similar to expert systems. One way that Artificial Intelligence approaches are argued to be able to reflect real-world processes is in dealing with uncertainty, particularly by using fuzzy logic (Zadeh, 1979). In this concept, rules or facts are not defined with binary precision, but rather values or distributions are assigned to indicate the degree or extent to which a description or factor applies, e.g., instead of defining tall as over 1.70 m (thereby consigning those 1.70 m or under to be not-tall or short), fuzzy logic permits heights to be assigned tallness quotients, representing the extent to which they conform to the adjective "tall" - 1.7 m might be 0.7, whereas 2.00 m would be more like 0.95. Systems using this can combine such uncertain or fuzzy variables and relationships into their reasoning.

However system dynamics modellers are well used to applying multipliers from table functions to represent similar fuzzy concepts in their models. For example, a decision to expand plant capacity may be based on current conditions concerning expected profitability of the product, projected growth in demand, closeness of current plant utilization to desired maximum, availability of finance. Variables representing each of these factors can be converted into a quotient or multiplier and combined to give an overall propensity or desirability to expand which would control an expansion decision (probably through an intension-to-build backlog that would trigger at an expansion increment).

- What is the relationship between ES & SD?

In a number of important respects system dynamics models involve many of the essential elements of expert systems, more if the whole process of modelling is included:

- SD models comprise formulations capturing knowledge about the structure and decision-making processes of real-world systems.

- They are able to cope with deep and underlying knowledge as well as simple rules, and are generally robust.

- In terms of the comparatively transparent code of DYNAMO syntax and of causal-loop, influence and flow diagrams there is an ability to explain to users how results occur, behaviour is generated, and inputs/parameters fit in.

- If the "system" is allowed to include a system dynamics practitioner then he can help elicit system behaviour and understanding - even though he may not be an expert in the particular domain.

Indeed, in as far as many claimed expert systems do not encompass all of these either - they may only be simple rule chains, may deal with only trivial or superficial representations, may not have complete explanations of reasoning etc., - it could be argued that at least in terms of functionality, a well designed and implemented system dynamics model could fit the bill just as well. It would probably also be much cheaper to implement for any large (complex) real-world system, especially for systems of the "practitioner assistant" rather than "expert consultant" type.

In the future this "competition" will become irrelevant. Already it has been recognised (Basden, 1984) that there needs to be a "blurring of the distinction between 'expert systems' and 'conventional' computing techniques, so that techniques are used according to their usefulness rather than their label". A hybrid model has already been reported by Levary & Lin (1988) called HESS which embeds a software lifecycle simulation model written in DYNAMO within a system which includes an input expert system to check compatibility of input vector, an output expert system which makes recommendations regarding the software development process, and a knowledge base management system which logs and reconciles input vectors with recommendations. Future generations of computers, particularly parallel processors, offer the likelihood of full integration between these technologies.

Basden A. (1984) "On the Application of Expert Systems" in Developments in Expert Systems, ed M. J. Coombs, Academic Press.

Clifford J., Jarke M., Lucas H. C. (1986) "Designing Expert Systems in a Business Environment" in Artificial Intelligence in Economics and Management, ed. L. F. Pau, Elsevier Science Publishers BV (North-Holland).

Coombs M. & Atty J. (1984). "Expert Systems: An Alternate Paradigm" in Developments in Expert Systems, ed. M. J. Coombs, Academic Press.

Cullen J. & Bryman A. (1988). "The Knowledge Acquisition Bottleneck: Time for re-assessment", Expert Systems, Aug., Vol 5, No. 3.

Hampshire N. (1988). "Introducing Declarative Languages". EXE Magazine, October, Vol 3, Issue 5.

Hertz D. B. (1988). The Expert Executive, John Wiley & Son.

Hudlicka E. (1988). "Construction and Use of a Causal Model for Diagnosis." Int. J. of Intelligent Systems, Vol 3, No. 3.

Levary R. R. & Lin, Chi Y (1988). "Hybrid Expert Simulation System (HESS)", Expert Systems, May, Vol. 5, No. 2.

Towris J. "Expert Systems in Freight Management." SERC Workshop on Expert Systems in Transport, Univ. of Leeds/SERC.

Waterman D. A. (1986). A Guide to Expert Systems, Addison-Wesley.

Winch G. W. (1984). "The Simulation Process and Techniques" in Policy Evaluation using REVEAL, ed. M. Small, ICL Publications.

Zadeh L.A. (1979). "A Theory of Approximate Reasoning". Machine Intelligence, Vol 9.