

8-16-1996

The Need for "Scientific" Information Systems

Frank H. Gregory

City University of Hong Kong, isfrank@cityu.edu.hk

Follow this and additional works at: <http://aisel.aisnet.org/amcis1996>

Recommended Citation

Gregory, Frank H., "The Need for "Scientific" Information Systems" (1996). *AMCIS 1996 Proceedings*. 287.
<http://aisel.aisnet.org/amcis1996/287>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 1996 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

The Need for "Scientific" Information Systems

[Frank H. Gregory](#)

City University of Hong Kong

Email: isfrank@cityu.edu.hk

The Principle of Falsification

Of all the philosophical principles that underpin our modern view of the nature of reality probably the most important is David Hume's distinction between "*Relations of Ideas* and *Matters of Fact*" (Hume, 1777). This is the foundation of British Empiricism from which Anglo-American analytical philosophy evolved. It replaced Descartes' rationalist thesis that knowledge of the physical world could be obtained by deduction from indubitable premises. It gave rise to the distinction between logical truth and factual truth.

In the following it will be argued that the recognition of the distinction between logical and factual truth is vital in information system design, yet it is a distinction that is not found in the information system design methodologies. Nor do the vast majority of information system designers recognise the need for any such distinction.

Most analytical philosophers would agree that there is a difference between logical and factual truth but how this distinction is to be made is still open to some dispute. The following is one plausible account.

Factual truth is the property of propositions that correspond to real world states of affairs. Logical truths are the constitutive rules of languages, formal and informal, used to express factual truths. Logical truths are invented, factual truths are discovered. Logical truths are necessarily true, factual truths are contingently true. The negation of a logical truth is a self-contradiction. The negation of a factual truth will be false but it will never be a self-contradiction. For those who are unfamiliar with this an example might help to make it clear.

"All crows are black" would generally be taken to be a factual universal. It can be falsified by establishing the truth of a particular statement that contradicts it. An example would be "Jack is a crow and Jack is white".

"All bachelors are unmarried men" is generally taken to be a logical universal. Here any particular statement that contradicts it, such as "Jack is a bachelor and Jack is a married man", is self-contradictory and cannot be true. "All bachelors are unmarried men" is a rule of English. "Jack is a bachelor and Jack is a married man" is a statement intended to be in English but it breaks a rule of English and is, therefore, self-contradictory.

The scientific principle of falsification follows from this. For any statement, p, about a real world state of affairs there must be another statement, q, where q contradicts p but q is not self-contradictory. This principle is inherent in Hume but today it is most closely associated with the name of Karl Popper (1992), who made falsification the main work-horse in his philosophy of science. For him factual universals were the same as scientific hypothesis, the job of a scientist was to generate hypotheses and test them by searching for particulars that would falsify them.

Programs can be written that act in accordance with the principle of falsification. An example is this simple program in PROLOG:

black (X) if crow (X).

incorrect_hypothesis (all_crows_are_black) if crow (X) and white (X).

If we add to this program the particular fact that Sam is a crow "crow (sam)" then ask for a list of black things "Goal: black (X)" the program will tell us that Sam is black "X = sam". If we add the particular fact that Peter is a crow and Peter is white "crow (peter). and white (peter)." then ask the program for a list of falsified universals "Goal: incorrect_hypothesis (X)" it will tell us that all crows are black has been falsified.

The Problem with Information Systems

The programs in most information systems do not incorporate the principle of falsification or any equivalent mechanism. The program rules in most systems are hard and fast. Particular facts, in the form of data, that contradict the rules are either ignored by the system or cannot be entered. These systems, therefore, operate with only particular facts and non-falsifiable universals. This gives three possibilities.

Firstly, it might be intended by the designers to have a system where the rules are all logically true and not about real world states of affairs. It might be argued that this configuration is appropriate to some information systems, say, those concerned with law. We can call this type of system "non-scientific".

Secondly, the computer system might comprise only logically true universals but be part of a larger system that contains factual universals. We can call these types of system "scientific".

Thirdly, the program contains only non-falsifiable rules and some of these are intended to generate data about states of affairs in the real world. We can call these types of system "unscientific".

Unfortunately, many, if not most of the information systems in operation today are unscientific. The traditional methodologies used to design them, such as Information Engineering and Structured Methods make no provision for the inclusion of falsifiable universals in the software. Nor do they make provision for the design of a wider organisational system in which the rules inherent in the software can be falsified.

The rules of information systems concerned with real world events need to be open to change because our knowledge of world changes. We might have good reason to believe that all crows are black until a rare species of white crow is discovered in a remote part of the world. More importantly the world itself changes. It may be true that all crows are black when build our system but then a mutant type of white crow develops and flourishes.

The methodology enthusiasts defend themselves by saying that their systems work in practice. The point is that they have, in principle, no way of knowing whether they work or not. The systems could be continuously outputting erroneous data. When mismatch between reality and data output is detected it is inevitably by the unsolicited action of users or customers, in other words, by accident. It never seems to occur to methodologists that there might be a plethora of undetected errors in their systems.

Possible Solutions

Although executable scientific programs, such as the one above, are possible the building of scientific information systems present a considerable challenge. The grammar of natural languages gives no indication of the difference between logically true and factually true statements. Most people are unconscious of the difference. Also the logical status of a statement can be regarded differently by different people and at different times. For us "all men are mortal" will generally be taken to be factual and open to falsification by finding a man who seems likely to live forever. But for the ancient Greeks any man-like being that was immortal was a god not a man. For them "all men are mortal" was a logical truth.

Knowledge based systems with learning capability qualify as scientific information systems. However, most of these, such as medical diagnostic systems, are built within the framework of long established disciplines. Such disciplines have a rigorously defined terminology. Given this the difference between

logical and factual truth is comparatively simple to make. Rigorous and uniform terminology will not normally be found in most private and public sector organisations. For example, the meaning of "satisfied customer" differs widely between organisations - in some it is defined as an absence of complains while in others it is defined as repeated business.

Analysts/designers could make their own arbitrary distinction between logical and factual truth in their information system designs. Unfortunately this would be likely to produce a mismatch between what the system output means and what the users think it means. For example, the system might show that 90% of customers were satisfied, meaning that 90% had not complained; but the users might take this to mean that 90% of customer had repeated business with the company. In this scenario the users would have to learn what the information provided by the system means. This would place an enormous burden on the users and poor results in user acceptance and operational efficiency should be expected.

A second possibility is for the analyst/designers to elicit the difference between logical and factual truth that is implicit in the users' language and knowledge. Knowledge elicitation by means of knowledge representation schema is well established. (Ringland & Duce, 1988) contains comprehensive account of the diverse schema. Unfortunately none of these schema are capable of representing the distinction between logical and factual truth.

The third possibility is for the users themselves to make the distinction between logical and factual truth. It is at this point that the British client led design methods, especially Soft Systems Methodology (SSM), start to become relevant. The SSM conceptual models of human activity systems can be understood as providing the rules of a language to describe the area of concern - a set of definitions of key terms (Gregory, 1993a).

SSM has had two uses. One as a general problem structuring method (Checkland & Scholes, 1990), the other as a front end to information system design (Wilson, 1990. Avison & Wood-Harper, 1990). In the latter context, the authors may claim, with some justification, that the client involvement has helped them to build the right system. Unfortunately it has not helped them to build the system right. The systems that have been built using SSM have been transaction processing systems that share the same faults as those built using traditional methodologies. These systems do not contain falsifiable universals - they do not realise the full potential of the SSM conceptual models. (Gregory, 1993b).

Logico-linguistic Modelling

SSM models are of the bubble diagram form. They comprise words in bubbles that are connected by arrows. The arrows represent "logical dependency" and in terms of formal logic can be translated as implication. A single logical connective is not sufficient to represent simple causal sequences let alone scientific information system. However, enhancements to the SSM models have been developed which give them the full power of modal predicate logic.

These logico-linguistic models can express the clients' vocabulary in the rigorous structure of modal predicate logic. They can supply all the logically true universals needed to create the sort of PROLOG program given above. They can also be used as framework for knowledge elicitation in which the distinction between logical and factual truth is unambiguous.(Gregory, 1993c, 1995).

Conclusions

The difficulties involved in designing scientific information systems are by no means insurmountable. In the forgoing, one method has been suggested but there are, no doubt, many other ways of doing it. The main problem is that people in the industry do not perceive the need to build scientific systems.

Organisational and user requirements are the concern of the management side of information system design. Management scientists usually have little knowledge of the sort of logic needed to give a formal specification. Implementation is left to software engineers whose main concern is the completeness and

consistence of their systems rather than the correspondence of data output with real world facts. What is needed is a logical and scientific structure throughout the analysis and design process.

References

Avison, D. E. & Wood-Harper, A. T. *Multiview: An Exploration in Information Systems Development*. Blackwell Scientific Publications, Oxford. 1990.

Checkland, P. B. & Scholes, J. *Soft Systems Methodology in Action*. Wiley, Chichester. 1990.

Gregory, Frank H. "Soft systems methodology to information systems: a Wittgensteinian approach." *Journal of Information Systems* 1993a Vol. 3, 149 - 168.

Gregory, Frank H. Logic and Meaning in Conceptual Models: Implications for Information System design. *Systemist*. 1993b, vol 15 (1).

Gregory, Frank H. "Cause, Effect, Efficiency and Soft Systems Models". *Journal of the Operational Research Society*, 1993c. Vol. 44, No. 4.

Gregory, Frank H. "Soft Systems Models for Knowledge Elicitation and Representation." *Journal of the Operational Research Society*, 1995, Vol. 46, No. 5.

Hume, David. *Enquiries concerning the human understanding and concerning the principles of morals*. 1777.

Popper, Karl. *The logic of scientific discovery*. Routledge, London. 1992.

Ringland, G. A. & Duce, D. A, Eds. *Approaches to Knowledge Representation*. Research Studies Press. Tauton, 1988.

Wilson, Brian. *Systems: Concepts, Methodologies and applications, Second Edition*. Wiley, Chichester. 1990.