A categorical logic for information systems

Michael Johnson

School of Mathematics and Computing, Macquarie University
AUSTRALIA
`email: mike@mpce.mq.edu.au`

Abstract

Category theoretic models of information systems have by now been widely used in industrial consultancies (for some examples see [6] and [3]) and the models themselves have been developed in some detail in, for example, [4], [1], [10], [5], [11], [2] and [8].

Although many information systems are never meant to be implemented computationally (for example paper-based enterprise wide information models are frequently used by senior executives to help understand and plan the development of large corporations), the *logic* of an information system corresponds to the logic of the query language that might be used for interrogating the system (whether or not it is implemented). Typically this logic corresponds to a variation of predicate logic applied to simple finite domains.

Most categorical information system modellers either avoid being explicit about the logic of their systems (thus acquiring the logic of whatever query language happens to be present in an implementation) or import the logic of topos theory [7] into their categorical models. The author argued in [4] that the logic of an information system should be determined by the categorical operations which can be performed within the system, and this is usually less than the logic provided by topos theory.

This paper analyses the information system specification process to determine precisely which categorical operations are necessary and then develops the logic internal to a category with the required operations. It turns out that the categories closed under these operations have a particularly well-behaved logic, but are not toposes, nor pretoposes, nor logical categories. We develop the theory of these categories in some detail.

We show using examples from industry that the "classifying category" of an information system must include

- A terminal object
- Pullbacks
- Finite coproducts, and
- A proper factorisation system
- A subobject classifier.

Furthermore, we show that coproducts must be disjoint and universal and that epis must be stable under pullback.

A category with the above properties has been called a *tos*, so named because it is intended to have as far as possible the same logical properties as a topos, but without requiring power objects (hence the deletion of "po" from topos). We develop the internal logic of a tos in the usual manner (see for example [9] or [7]). The only particular need for care here is in the definition of implication (since the usual definition requires power objects [7]) and in the use of universal quantification (since inverse image functors for tosses do not in general have right adjoints). Of course defining implication as the classifying map of the equalizer of conjunction and the first projection suffices in the first case. Universal quantification on the other hand needs to be restricted.

It might be hoped that a tos provides a closer representation of the operations available within an information system because industrial applications generally make no use of power objects and do not need general colimits (which are frequently computationally expensive). It is easy to see that tosses *are* strictly more general than toposes because there are examples which do not have power objects. However, at present the question of whether tosses happen to be finitely cocomplete remains open.

References

[1] P. Buneman, A. Jung and A. Ohori, Using powerdomains to generalise relational databases, *Theoretical Computer Science*, **91** (1991), 23–55.

[2] B. Cadish and Z. Diskin, Algebraic graph based approach to management of multibase systems 1: Schema integration via sketches and equations, *Next Generation of Information Technologies and Systems* (NGITS'95), Naharia, Israel, (1995), 69–79.

[3] C.N.G. Dampney, M. Johnson and P. Deuble, Taming large complex information systems, in D. Green and T. Bossomaier (eds) *Complex Systems*, IOS Press, (1993), 210–222.

[4] C.N.G. Dampney, M. Johnson and G.P. Monro, An illustrated mathematical foundation for ERA, in C.M.I. Rattray and R.G. Clarke (eds) *The unified computation laboratory*, Oxford University Press, (1992), 77–84.

[5] A. Islam and W. Phoa, Categorical models of relational databases I: Fibrational formulation, schema integration, preprint (1993).

[6] M. Johnson and C.N.G. Dampney, On the value of commutative diagrams in information modelling, *Springer Workshops in Computing* (1994), 47–60.

[7] S. Mac Lane and I. Moerdijk, *Sheaves in geometry and logic*, Springer-Verlag (1992).

[8] F. Piessens and E. Steegmans, Categorical data specifications, *Theory and Applications of Categories*, **1** (1995), 156–173.

[9] A.M. Pitts, Categorical logic, in S. Abramsky, D.M. Gabbay and T.S.E Maibaum (eds) *Handbook of Logic in Computer Science*, VI, Oxford University Press (1996).

[10] R. Rosebrugh and R.J. Wood, Relational databases and indexed categories, *Canadian Mathematical Society Conference Proceedings* **13** (1992), 391–407.

[11] S. Vickers, Geometric theories and databases, *London Mathematical Society Lecture Notes* **177** (1992), 288–314.