

# Towards a Unified Algebraic View of Structured Systems Development Models

T.H. Tse  
London School of Economics  
University of London

## Research Problem

Various models have been proposed under the name of structured systems development. They are widely accepted by practising systems developers based on their simplicity of use and ease of communication. Because of the lack of a theoretical framework, however, transition from one model to another is arbitrary and can only be done manually. Automatic development aids tend to be ad hoc and model-dependent. There is a need, therefore, to provide a unified framework for the structured models.

## Background Literature

A brief survey of the literature indicates that there are quite a number of structured models in common use. Examples are DeMarco data flow diagrams [3, 4, 14], Jackson structure diagrams [7], Warnier/Orr diagrams [10], and Yourdon structure charts [15]. It has been found that different structured models are suitable for different situations depending on the environment [13], emphasis [2], and stage of development [8]. Although transformations from one model to another are often recommended by authors, they are only done manually. To solve this problem, a review has been made on the formal models used in computer science and information systems. Models such as Petri nets [12], general net theory [11], category theory [9], and initial algebras [6], have been studied. The last approach seems the most suitable for defining a formal view of the structured models. It has a rich mathematical linkage with category and algebraic theories. But at the same time, the concepts can be stated in simple terms for systems designers who do not want to be involved with complex theories. It is much easier to validate the specifications or prove the correctness of programs through the algebraic formalism. Interpreters for algebraic specifications such as OBJ [5] and Clear [1] are already available.

## Research Design

An algebra is a family of objects that satisfy a formal structure called a signature. Given a system, we may use different specification methods to describe the processes and their relationships. Likewise we can define different algebras over the same signature. All these algebras can be linked up using homomorphisms, or signature-preserving functions. The homomorphisms enable a systems designer to forget about minor syntactical differences between individual specification methods, so that he can concentrate on the major issues in a system. The algebra which has the richest context is called an initial algebra, denoted by  $T_{\Sigma}$ .

It has the property that, given any other algebra to  $A$  over the same signature, there exists a unique homomorphism mapping  $T_{\Sigma}$  to  $A$ . Because of the guaranteed homomorphisms, the initial algebra can be used to link up various structured models. An initial algebra will be defined and linked to DeMarco data flow diagrams, Jackson structure text, and Yourdon structure charts through homomorphisms and equations. A prototype system to implement the algebra is being studied. It will run on a Sun workstation under Unix.

## Contribution of This Research

As a result of the research, structured specifications can be transformed from one form to another through homomorphisms and equations. The most suitable model can be chosen for a target system independent of user familiarity. Algebraic interpreters may be adapted to validate the specifications. Automatic development aids for one methodology may be applied to another through transformations.

## References

- [1] R.M. Burstall and J.A. Goguen, “The semantics of Clear, a specification language”, in *Abstract Software Specifications: Proceedings of the 1979 Copenhagen Winter School*, D. Bjorner (ed.), Lecture Notes in Computer Science, vol. 86, Springer, Berlin, Germany, pp. 292–332 (1980).
- [2] M.A. Colter, “Evolution of the structured methodologies”, in *Advanced System Development / Feasibility Techniques*, J.D. Couger, M.A. Colter, and R.W. Knapp (eds.), John Wiley, New York, NY, pp. 73–96 (1982).
- [3] T. DeMarco, *Structured Analysis and System Specification*, Yourdon Press Computing Series, Prentice Hall, Englewood Cliffs, NJ (1979).
- [4] C.P. Gane and T. Sarson, *Structured Systems Analysis: Tools and Techniques*, Prentice Hall, Englewood Cliffs, NJ (1979).
- [5] J.A. Goguen and J.J. Tardo, “An introduction to OBJ: A language for writing and testing formal algebraic program specifications”, in *Software Specification Techniques*, N.H. Gehani and A.D. McGettrick (eds.), Addison-Wesley, Reading, MA, pp. 391–420 (1986).
- [6] J.A. Goguen, J.W. Thatcher, E.G. Wagner, and J.B. Wright, “Initial algebra semantics and continuous algebras”, *Journal of the ACM* **24** (1): 68–95 (1977).
- [7] M.A. Jackson, *Principles of Program Design*, Academic Press, London, UK (1975).
- [8] R.J. Lauber, “Development support systems”, *IEEE Computer* **15** (5): 36–46 (1982).
- [9] S. Mac Lane, *Categories for the Working Mathematician*, Springer, New York, NY (1971).
- [10] K.T. Orr, *Structured Systems Development*, Yourdon Press Computing Series, Prentice Hall, Englewood Cliffs, NJ (1977).
- [11] C.A. Petri, “Introduction to general net theory”, in *Net Theory and Applications*, Lecture Notes in Computer Science, vol. 84, pp. 1–19 (1980).

- [12] W. Reisig, *Petri Nets: An Introduction*, EATCS Monographs on Theoretical Computer Science, vol. 4, Springer, Berlin, Germany (1985).
- [13] O. Shigo, K. Iwamoto, and S. Fujibayashi, “A software design system based on a unified design methodology”, *Journal of Information Processing* **3** (3): 186–196 (1980).
- [14] V. Weinberg, *Structured Analysis*, Yourdon Press Computing Series, Prentice Hall, Englewood Cliffs, NJ (1980).
- [15] E. Yourdon and L.L. Constantine, *Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design*, Yourdon Press Computing Series, Prentice Hall, Englewood Cliffs, NJ (1979).