COMBINING GENERIC STRUCTURES AND SYSTEMS ENGINEERING TO MANAGE COMPLEXITY IN SYSTEM DYNAMICS MODELLING

Alan C. McLucas and Michael J. Ryan[1]

Abstract. Expert system dynamicists are those who have developed the skills to perceive *structure*—that is, they have ability to recognise that dynamics appearing to be very different on the surface are actually caused by fundamentally similar mechanisms. They then use these skills very effectively to build models of complex problems. Desire to exploit knowledge of fundamental structures in system dynamics models has led to the formulation of *molecules* of system dynamics structure. But there is more we can do to facilitate learning about, and recognition of, structure as well as improve system dynamics modelling methodology. This paper argues that aspects of systems engineering practice can be integrated with system dynamics to produce a methodology which exploits knowledge of structures, utilises top-down model formulation and bottom-up construction of models, thereby enabling management of the complexity encountered during model building. The proffered methodology enables all modellers, even the least experienced, to quickly and reliably build robust models of complex problems. How this is achieved is explained and demonstrated.

GENERIC STRUCTURES IN SYSTEM DYNAMICS MODELLING

In 'Industrial Dynamics', Jay W. Forrester (1961: 2) recognised the importance of guiding system dynamics students to studying principles of structure and causes underlying dynamic behaviour:

> *The rapid strides of professional progress [in system dynamics modelling] come when the structure and principles that integrate [synthesise (in systems engineering terms)] individual experiences can be identified and taught explicitly rather than by indirection and diffusion. The student can inherit an intellectual legacy from the past and build his own experience upward from that level rather than having to start over again at the point where his predecessors began.*

Paich (1985: 126-132) stresses an insight by Richmond that there could be considerable utility in isolating and defining 'atoms of structure', those 'primitive feedback loops, which generate behaviour of basic processes', which experienced system dynamicists recognise as important.

Since those formative times in the development of the system dynamics discipline, there have been a number of dedicated researchers and practitioners working to progressively identify and define these fundamental building blocks of structure. These building blocks have been variously referred to as 'atoms' (Richmond, 1977), 'molecules of system dynamics structure' (Hines, et al., 1996; 1997; 2000), 'common modules' (Coyle, 1996), 'organelles' (Malcszynski, 2005), and 'modules' (McLucas, 2005).

The term 'molecule' is used in the remainder of this paper to describe those building blocks of system dynamics structure defined by Hines et al. (1996; 1997; 2000) and McLucas (2005). The term module, which includes molecule, is used to describe the fundamental building blocks used during the process of building system dynamics models. Figure 1 illustrates how a molecule is defined in terms of a module boundary and inflows and outflows (McLucas, 2005).

Whilst the import and export to datasets are important for the overall operation of the model, and this has to be verified at some stage, the interfaces (for the purposes of integrating with other modules, sectors or models) are:

- physical inflows,
- physical outflows,
- information inflows, and
- information outflows.

Many researchers and practitioners have had thoughts on generic structures along the lines of Graham (1977), Forrester (1968), Goodman (1989), Paich (1985) and Hines, et al. (2000). Significant contribution has been made by Coyle (1996) who makes repeated reference to, and provides examples of, 'common modules' which he has defined and described both as sets of influence diagrams and corresponding functional blocks of code in COSMIC® software. It is emphasised at this point that these functional blocks are not the algebraic definitions commonly encountered in system dynamics software, such as STEP, PULSE, or Delay.

Perhaps the most familiar molecule of system dynamics structure is that defined by Sterman (2000: 266). This is the first-order linear positive feedback system, as depicted in Figure 2.

Depending on the exact definition of molecule used, it might be possible to define more than 50 functional building blocks (Hines, et al., 2000, define some 46 molecules).

MANAGING COMPLEXITY WITH SYSTEMS ENGINEERING

Traditional engineering design methods are based on a bottom-up approach in which known components are assembled into subsystems from which the system is constructed. The system is then tested for the desired properties and the design is modified in an iterative manner until the system meets the desired criteria. This approach is valid and extremely useful for relatively straightforward problems that are well defined. Unfortunately, complex problems cannot be solved with the bottom-up approach.

Systems engineering begins by addressing the complex system as a whole, which facilitates the initial allocation of

[1] School of Information Technology and Electrical Engineering, University of New South Wales, Australian Defence Force Academy, Northcott Drive, CANBERRA, ACT, 2600, Australia.