CHOICE AND DECISION-MAKING IN ENGINEERING AND ARCHITECTING OF COMPLEX SYSTEMS

Alan C. McLucas

Abstract. In this article it is argued that, far too often, conceptualisation of complex systems, development of user requirements and subsequent systems engineering activities do not produce the results expected. Attempts to more rigorously apply systems engineering practices fall short because the roles choice and decision making play are not well understood. Whilst systems engineers are taught to carefully weigh up options, the reality is that naturalistic decision-making, mental shortcuts or experience-based heuristics are routinely applied. So, systems architecting, which acknowledges the role of stakeholders and the use of heuristic-based decision making, is an important adjunct to conventional systems engineering, when it comes to development of complex systems. Conceptualisation of complex systems is more of an art than a science, and that art is affected in its application by organisational and cultural influences.

'Another factor in overruns and delays [in complex acquisitions] is uncertainty, the so-called unknowns and unknown unknowns. Uncertainty is highest during conceptualization, less in design, still less in redesign, and least in upgrade. As with complexity, the higher the level, the more important become experience-based architecting methods'[1]

INTRODUCTION

Extensive participative action research into computer-based information systems (CBIS) [2] concluded that not only can requirements not be made fully explicit at the start of a project, they cannot be made fully explicit at all. Soft systems modelling approaches such as the Soft Information Technologies Methodology (SISTeM) used by Atkinson [3] to aid development of CBIS and Iterative and Interactive Strategy Development [4] offer considerable promise as aids to systems architecting. These techniques and systems architecting have potential to fill significant gaps that currently exist in requirements engineering and systems engineering practices. The author of this article reached a similar conclusion to Sutton in respect of Defence communications and information systems, development of decision support systems for the dynamic management of Defence preparedness, and the management of Defence capability.

THE PROBLEM

Requirements engineering and management can be problematic for a number of reasons. Complexity of systems and rapid changes in technology are obvious ones. However, the most pervasive reasons are not technical in nature. They are organisational, cultural, or caused by failures in human cognition [5] and linguistic [2,5].

Maier and Rechtin [6] observe that different techniques are required in the engineering of systems at high levels of complexity than at low ones. Purely analytical techniques, powerful for lower levels, can be overwhelmed at higher ones. At higher levels architecting methods, experience-based heuristics, abstraction, and integrated modelling techniques must be called into play [1]. The basic idea behind systems engineering techniques is to simplify problem solving by concentrating on its essentials. Consolidate and simplify the objectives. Stay within the guidelines. Put to one side minor issues likely to be resolved by the resolution of major ones. Discard the nonessentials. Model (abstract) the system at as high a level as possible, then progressively reduce the level of abstraction. In short, **Simplify!**

Where quantitative models might be used, populating those models built at high levels of abstraction is a problem in itself because data is usually gathered at much lower levels and is not easily aggregated, or if aggregated is meaningless [7,5]. The necessary data is unlikely to be available for analysis of novel problems, or for development of novel systems.

The concept that a complex system can be progressively partitioned into smaller and simpler units—and hence into simpler problems—omits an inherent characteristic of complexity, the interrelationship among units, the strongly coupled nature of complex systems and systemic problems [5,6]. Poor aggregation and partitioning during development can actually increase the complexity, a phenomenon all too apparent in the organisation of work breakdown structures.

This primacy of complexity in system design helps explain why a single 'optimum' seldom if ever exists for such systems. There are just too many variables. Variables are a mix of 'hard' and 'soft' frequently strongly-coupled, there are too many stakeholders, and too many conflicting interests [5,6]. No practical way may exist for obtaining information critical in making the "best" choice among quite different alternatives.

WORKING DEFINITIONS

System. A system is a collection of different things that together produce results unachievable by themselves alone. The value added by systems is in the interrelationships of their elements. These interrelationships produce the emergent properties of systems, where the whole is greater than the sum of the parts.

Systems engineering. Systems engineering is the art and science of creating a product or service, based on phased efforts that involve definition, design, development, production, and maintenance activities. The resulting product or service is functional, reliable, of high quality, and trustworthy, and has been developed within cost and time constraints [8]. Expanded definitions are provided by MIL-STD-499A and MIL-STD-499B.