

The Wrong Kind of System Engineering

Fifth European Systems Engineering Conference

18-20 September 2006

Colin Brain, MSc, CEng, DipEM, MIMechE

SE Validation Limited, 10 College Street,

Salisbury, SPI 3AL, UK

cjb@sevalidation.com

Copyright © 2006 by Colin Brain Published and used by INCOSE with permission.

Abstract. Effective communication is important and often depends on establishing a common context. This paper looks at contexts for system engineering and demonstrates that there are hidden (business) assumptions in the way that they are often viewed. A working definition of system engineering is derived: “the application of system-related knowledge for the achievement of enterprise goals”, that links system engineering with business strategy. Since there is more than one type of strategy, the paper explores this relationship, providing information for system engineers working in inter disciplinary or multi-cultural environments .

It shows that a classical approach to system engineering fits with a classical business strategy, but that using this approach unchanged in other types of organisations may be wrong. The paper uses an established business model to suggest how the situation might be improved. It concludes that there is a need to focus more on the ends that system engineering seeks to achieve. By doing this we stand a much greater chance of not using the wrong kind of system engineering and of adding value in inter-disciplinary and multi-cultural environments.

INTRODUCTION

The theme of this conference is “Adding value in inter disciplinary and multi-cultural environments”. Nine years ago, at an UK conference on System Engineering for Defence (Anon, 1997) one industry speaker described system engineering as “The disciplined application of common sense to problems too big for one mind to contain”. Whilst few would totally agree with this definition, most can probably identify with its sentiments. However, this definition contains a clue as to why we need to do worry about system engineering in inter disciplinary and multi-cultural environments and why there might be a problem with the ‘The Wrong Kind of Systems Engineering’. This centres on the notions surrounding the term “common sense”. Common sense is not common, it is private, and is the product of culture, training and environment, particularly during the formative years. Thus the problem with common sense is that the assumptions that underlie it are hidden, including to the person invoking it. (Frank, 2006) suggests that “successful systems engineers constantly question information they are given”. Whilst this is undoubtedly true, I feel that there should be a codicil to this: “even more successful systems engineers constantly question the system engineering theories with which they work”. According to (Whittington, 2001) “*drawing on his work with American senior managers Argyris (1977) warns that nothing is more dangerous than to leave underlying assumptions hidden. Until we surface our implicit ‘theories of action’, we cannot test their accuracy and amend them to the conditions of the day. Those who do not actively confront their underlying assumptions are condemned to be ‘prisoners of their own theories’*”. This paper addresses one framework that may help system engineers to confront some of their underlying assumptions about system engineering. Of all the sub-disciplines of system engineering, one might expect that safety engineering was the one that contained the least number of implicit assumptions. Yet (Maguire et al, 2006) has shown that there are important differences in perception of the meaning of some of the most common words that form the foundation of communication on system safety issues – and this is within one culture and one language - English.

SYSTEM ENGINEERING

Definition. So what does the phrase systems engineering mean? Even within systems engineering and related disciplines there is still a debate on this. From its name, it is a form of engineering and one enduring definition for engineering is that it is: “the application of knowledge for the creation of wealth”. Here, it is understood that wealth means much more than pure financial gain. Based on this definition, one might define the application of systems engineering within an enterprise as: “the application of systems-related knowledge for the achievement of enterprise goals”. Putting this in a practical context, this could be interpreted within the European Business Excellence Model (BEM), for example, as using system knowledge to guide the development of the ‘Business Enablers’ (Leadership; People; Policy & Strategy; Partnerships & Resources; and Processes). Many would argue that there are more than enough systems engineering definitions in the world already without inventing a new one, and that it would be better to use a standard one, such as the INCOSE definition: “Systems Engineering is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem...”. However,

on analysis, this appears to be a definition of systems engineering in the context of developing hardware-centric systems using a waterfall development cycle. (Maier, 2006) discusses the problems that software engineers have with working with this kind of approach. Architectures may be more important than functions; the development approach may be spiral, or more 'bottom-up' than 'top-down'; the engineering may be through-life, not just development. Thus it is clear that even the INCOSE definition contains un-stated assumptions about its context of application. A simplistic definition has been adopted for this paper in an attempt to minimise the number of hidden assumptions. However, before we leave this topic, there is one further point to draw out from the INCOSE definition. It talks about "successful systems", but clearly a system itself does not have the capacity to be successful. A system can be said to be successful if it satisfies its stakeholders. Clearly these stakeholders will not just be within one enterprise and will not have identical goals, or contract negotiators and contract lawyers would be endangered species. So success in systems terms is, at best, likely to be satisficing - satisfying enough of a stakeholder's needs at a 'price' he or she finds acceptable. Here price implies more than money. The user may accept lower reliability, more maintenance, more procedural restrictions and a smaller operating envelope than they aspired to, because the new system is still better than its predecessor. The system realization enterprise may accept a lower profit than they forecast, since market conditions do not give them viable options for improving this. Customers, suppliers and Users all operate within larger systems. Indeed one could consider an enterprise as being a system designed to deliver the enterprise goals. The next section considers in more detail this link with business strategy, but an alternative way of expressing the aim of this paper is to link the architecture of the enterprise with the architecture of the enterprise's approach and means of utilising systems knowledge.

BUSINESS STRATEGY

Types of strategy. If system knowledge is going to support the achievement of enterprise goals through the 'business enablers' then it must both influence, and be influenced by, the enterprise's approach to business strategy. At least one management theory (Whittington, 2001) identifies four distinct approaches to business strategy, each one suited for a particular type of enterprise. The figure below shows how these are differentiated on two axes, outcomes and processes. Vertically the differentiation is whether there is a single desired outcome (usually profit) or multiple desired outcomes. Horizontally the difference is marked by whether the processes used to achieve the outcomes are mainly deliberate or emergent.

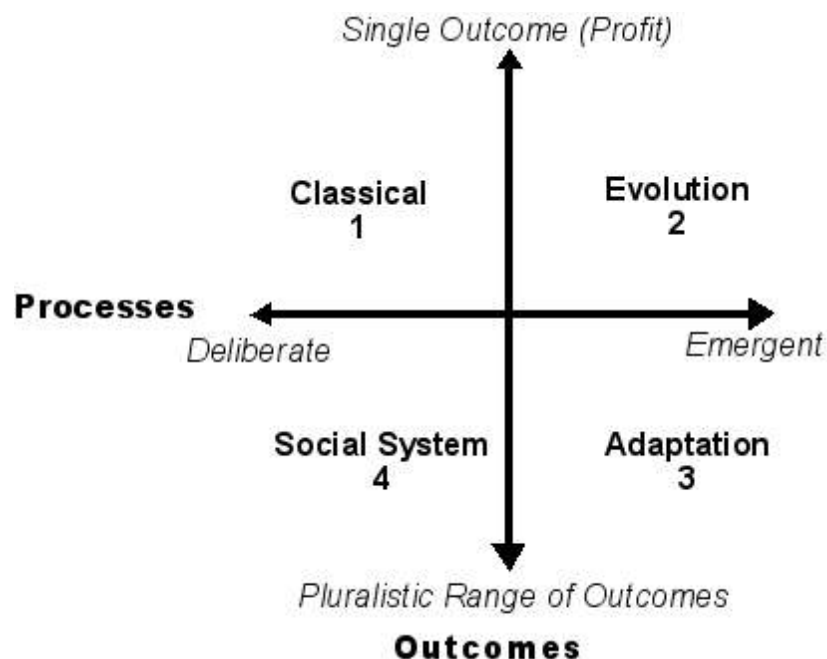


Figure 1. Showing four approaches to strategy

Classical. For the first 'classical' quadrant, the basic managerial assumption is that it is both possible and desirable to plan and invest for the long term. Enterprises operating in this quadrant are typically mature and capital intensive. Their maturity has led to the development of well-honed processes and their need for capital gives them a clear 'profit' focus to achieve adequate return on capital for their shareholders (at least in Western economies). The management gurus for this kind of approach are people such as Sloane, Ansoff, Drucker and Porter who recommend the adoption of planned, top-down 'rational' approaches to establishing simple and clear objectives. The underlying beliefs behind these recommendations include: that it is possible to foresee (and to some extent control) the immediate future; that formulation should be remote from, and precede, implementation; and that short-termism should be avoided in favour of longer-term planning. Classical

approaches to business strategy were in favour in the 1960s, the formative years of much of what we know as 'classical' system engineering. Thus it is not surprising that both of them take a very 'rational' view - analysis first, planning second and top-down-managed implementation third. Both the 'guru' recommendations and their foundation beliefs are entirely consistent with 'system engineering' as it is widely recognised, suggesting that it fits well in this type of business, for example, a large prime-contractor organisation. The metric most likely to be used for the effectiveness of 'system engineering' in this type of enterprise is contribution towards the bottom line – the business outcome sought.

Evolution. Quadrant 2 is heavily market-driven and until recently it was not obvious that enterprises in this quadrant would wish to adopt any 'system-based' approaches. They share with quadrant 1 the focus on profit, but usually do not invest in developing people and processes for the long-term on the basis that markets are too unpredictable to warrant this type of analysis and investment. Instead they concentrate on keeping their costs low and their options open. Rather than return on capital, their focus is on survival, market share and cost of sales. The relevant companies in this sector tend to be emerging players in the systems field and also Anglo-Saxon conglomerates who follow a hire and fire route to keeping their fixed costs down, often through the use of contract staff or services. Thus the processes they use are emergent and largely dependant in system engineering terms on the staff or organisations they hire. Their approaches are likely to be amalgams of those of the contractors they use. If they choose well, the specialist approaches may be closer to the 'leading-edge' than those of the companies in the first quadrant, but there is more of a danger that such approaches will not be well integrated and fully-compatible. In this type of organisation, senior managers frequently lack the capability to plan remotely from those implementing the system engineering, but this can be beneficial by allowing increased flexibility where resources are competent and not unduly constrained. The Evolution (survival) approach to business strategy came to the fore in the 1980s.

Adaptation. (Whittington 2001) calls this quadrant Processual, which is not a term that will be familiar to many system engineers, hence the revised name. Adaptation approaches are important to system engineering since this is where much of the system engineering innovation and problem solving comes from – 'the growing buds'. One can only employ deliberate processes once these have been properly established - until then they are emergent. The underlying beliefs of the management gurus who advocate this kind of approach, such as Simon, Mintzberg, Pettigrew, Cyert and March, are that strategy should follow implementation and should be established through a continuous crafting process, as a result of patterns (knowledge) emerging from small steps. Thus new problems, such as engineering network centric systems, cannot be adequately dealt with in quadrants 1 and 4, because the required deliberate processes have not yet been established. Quadrant 2 companies are generally not interested in investing in new competencies, but are likely to hire quadrant 3 companies to do this for them. Thus quadrant 3 is where most knowledge-based businesses are likely to operate and these are mainly trading on intellectual capital, not financial capital or market ratings. However, intellectual capital is largely people based, so it is not surprising that this sector seeks multiple outcomes, rather than a single profit motive. Technical excellence, leading-edge innovation, scientific freedom, or 'lifestyle' employment may all be just as important to the key knowledge resources. So organisations in this sector are unlikely to be able to successfully pursue a single profit agenda because of their needs to support effective recruitment and retention. Implementing in new engineering contexts where there is a process vacuum places considerable demands on the competence of the individuals involved. It is not surprising that this is the only quadrant where the gurus suggest that core competence is of prime importance. However, it is often difficult for an 'adaptation' organisation to maintain leading edge core competence right across system engineering and its sub-disciplines in any particular domain. It is much more likely that organisations will specialise, leading to a shortage of emergent capability at the (higher) system and system-of-systems levels. Adaptation strategies were initially a product of the 1970s.

Social System. (Whittington 2001) calls these Systemic strategy approaches. These are the most recent of the four and share the same use of deliberate processes with classical strategies, but are differentiated by the fact that the enterprise is seeking to achieve pluralistic outcomes. Very often the organisation's concentration is on managing within multiple time-varying constraints; perhaps political, staff numbers, financial, policy, etc. Typical enterprises that fall into this category are government departments, agencies and QANGOs, whose involvement in system engineering is often as customer, or stakeholder. The gurus for this type of organisation include: Swedberg, Whittington, Whitley, Hu and Pascale and the typical beliefs associated are that: there is not necessarily a single universal model; rational planning is possible, but only within specific social and cultural contexts; decision makers are embedded in densely woven social systems; strategic goals and processes (e.g. system engineering) should reflect the local social and economic systems in which they are implemented; and the external world has the biggest impact on strategy. This leads to the recommendation that planning should focus on the short and medium term and must be done with cognisance of the influencing social and economic factors, ideally in a distributed way through people embedded within the organisation. In 'classical' (quadrant 1) system engineering terms, these recommendations conflict with the desire: to have a single system architect; to have planning done remotely based solely on technical considerations; and to adopt a consistent top-down approach. Thus system knowledge is unlikely to be applied effectively in the social-system type of enterprise by using the 'classical' system engineering model, but might be applied more effectively through, say: 'middle-out', 'soft systems' and 'system-of-system' approaches. The need for responsiveness to short-term pressures also drives the need to manage requirements in a much more agile way than is classically the norm. Thus there could

potentially be a better fit between quadrant 4 ‘customers’ and quadrant 2 ‘primes’ using quadrant 3 ‘experts’ than with quadrant 1 ‘primes’ with self-contained organisations.

SYSTEM ENGINEERING METHODS

Solution engineering. In a presentation on system engineering that the author attended many years ago, it was stressed that the fundamental starting point for system engineering was identifying the customer. The distinguishing characteristics of a customer were said to be that they had: “a requirement in one hand and a corresponding bag of gold in the other”. Within the frame of reference established here, ‘classical’ system engineering methods represent the means of satisfying the requirement and thereby earning the bag of gold to pay for the capital employed. Naturally if the customer changes the requirement then he needs an additional bag of gold. This amounts to ‘solution’ system engineering.

Problem engineering. System engineering that starts with a customer patently does nothing to address how the customer gets the requirement or the corresponding bag-of-gold. Unless of course one assumes that he in turn has a customer with a bigger requirement and a bigger bag-of-gold, which seems to be a common, but usually unfounded, assumption. However, ‘social’ system engineering for the customer needs to consider where both that requirement and that bag-of-gold come from and how system knowledge can be used to optimize both. This could be called ‘problem’ system engineering to differentiate it from the classical solution-focus. Problem system engineering should thus aim to reconcile the pressures of pluralistic time-variant goals and support distilling these into clearly-stated requirements that are reasonably stable in the medium term. It should also excel at managing the gap between evolving user requirements and fixed contract specifications.

Complexity. There are many different complexity indicators used across science and engineering. For system engineering, complexity tends to come from the number of interfaces or interrelationships that need to be managed. For quite a number of years I have used several indicators of complexity when considering how to approach a task (Brain, 2004). A task is complex if there is one, or more, of the following is present:

- Multiple system architects and/or groups of stakeholders;
- Different component system lifecycles (often involving the integration of systems that were not designed for integration);
- Systems that are difficult to partition, particularly with respect to:
 - Non-functional requirements;
 - Service-based architectures;
 - Software and people within the system.

All these can be seen to be related to the four quadrants outlined above, although the drive may be 'bottom-up' rather than 'top-down'. Multiple system architects or groups of stakeholders imply a plural set of desired outcomes, which in turn implies the need for adaptation or social-system approaches. Different lifecycles also imply that the systems will not be designed at one time and with one architect and this undermines the 'classical' need to be able to reasonably foresee the future within the timescales involved in system realization. Systems that are difficult to partition reduce the engineer's ability to plan remotely from implementation, probably involve coupling between several levels of any system hierarchy and span multiple customer/supplier relationships. Thus these subjective complexity indicators could be seen as masking objective criteria about the applicability of classical system engineering methods. As indicated above, many of these complex system issues involve as much, or more, reasoning about the problem context as they do about the realization of the solution. There is a large overlap between ‘problem’ and ‘complex’ system engineering, so it is natural to expect that the methods of both would be similar. The table below presents a number of aspects of the need for ‘problem’ engineering methods, with recommendations of how these might be satisfied.

Need	Aspects	Recommended approaches
Communication	<ul style="list-style-type: none"> • Problem/solution interfaces • Communication of concepts • Agreeing problem architecture (rather than solution architecture) • Communicating pluralistic goals 	<ul style="list-style-type: none"> • Domain terminology (Jackson, 1995) • Synthetic environments • Problem frames (Jackson, 2001) • Goal Structuring Notation
Analysis	<ul style="list-style-type: none"> • Support for distributed analysis • Socio-technical systems 	<ul style="list-style-type: none"> • Problem and solution patterns • ‘Holonic’ methods (Koestler, 1967) • Matching designer and user expectations
Implementation	<ul style="list-style-type: none"> • Work and task breakdown 	<ul style="list-style-type: none"> • Decision-driven methodologies
Validation	<ul style="list-style-type: none"> • Maintaining through-life verification and validation 	<ul style="list-style-type: none"> • Assurance cases based on Claim / Argument / Evidence networks

Table 1. Complex/problem engineering methods

CONCLUSION

This short paper has suggested a minimalist definition for system engineering in an enterprise and linked this to the achievement of enterprise goals. By considering different approaches to strategy for different organizational types it has argued that 'classical' system engineering is a good fit to the 'classical' organization. However classical system engineering does not fit so well for other very important organization types. The fundamental beliefs that underlie these types of organization, and the recommended approaches for success, appear incompatible with what is usually recognized as system engineering. Thus they warrant different formulations of system knowledge if this is to help achieve their (plural) business goals.

For this to happen, system engineering needs to first consider the ends before it prescribes the means. Just as a professional engineer's first question should always be: "Am I competent to do this task?" so the first question for a system engineer should be "What formulations of system knowledge do I have to achieve the goals for this system and are these adequate?"

Distinctions were drawn between classical, evolution, adaptation and social-system business models and these were addressed from the perspective of system engineering. A particular separation was made between solution and problem engineering and this was related to measures of complexity. Using these frames of reference, the paper suggested some system engineering methods that may be more suitable to enterprises with pluralistic goals, emphasizing the fundamental need for communication and to be able to reason about the problem, rather than just the solution.

Good communication relies on common understanding of the contexts in which others are working. This paper has sought to expose that there are multiple contexts for system engineering and that simply assuming that these are a matter of 'common sense' is unhelpful. If we as a community start to focus more on the goals, problems or outcomes for which we are employing system knowledge, we can ensure that we are not advocates for the wrong kind of system engineering and that we are adding value in inter-disciplinary and multi-cultural environments.

REFERENCES

- Anon., "Proceedings of the 1997 DERA/RMCS Symposium on "System Engineering for Defence"." (Shrivenham October 1997), p 17.
- Brain C.J., "The Validation of Complex Systems", Coping with Complexity Workshop, (Bath University , September 16-17 2004). <http://www.cs.bath.ac.uk/~complex/cwc2004/program.htm>
- Frank, M., "Knowledge, Abilities, Cognitive Characteristics and Behavioural Competences of Engineers with High Capacity for Engineering Systems Thinking (CEST)." *Systems Engineering* Vol 9, No 2, Summer 2006, p 101.
- Maguire, R.L & Brain, C.J., "History and Perception of the Language Used in the Safety Domain." The 1st International Conference on System Safety Engineering, (IEE London, June 6-8, 2006).
- Maier, M.W., "System and Software Architecture Reconciliation", *Systems Engineering* Vol 9, No 2, Summer 2006, p 149.
- Jackson, M., *Software Requirements & Specifications*, Addison Wesley, 1995.
- Jackson, M., *Problem Frames*, Addison Wesley, 2001.
- Koestler, A., *The Ghost in the Machine*, Hutchinson, 1967.
- Whittington, R., *What is Strategy – and why does it matter?*, 2nd edition, Thompson, 2001.

BIOGRAPHY

During a six-year engineering apprenticeship at the DHAETS, Colin gained a first-class honours degree in aeronautical engineering. This was followed by an MSc in Aircraft Design at Cranfield. After five more years working on civil aircraft, Colin moved to the A&AEE, Boscombe Down to evaluate military aircraft engineering systems. During 30 years at Boscombe Down the organisation transitioned from central government to the private sector and Colin occupied a number of change-management roles, following the award of a Diploma in Engineering Management. One of Colin's last roles, at what was by now part of QinetiQ, was the development and management of the System Assurance capability and during this time he was made a QinetiQ Fellow. Retiring early from QinetiQ, Colin established the specialist company SE Validation Limited in 2004.