Enterprise and the Technology Environment

Peter Brook
Dashwood Consulting
Dashwood House, Manby Road
Malvern, Worcs WR14 3BB
UK

dashwoodbrook@aol.com

ABSTRACT

This paper discusses the nature of enterprises, both in the business and military domains, and puts forward a model for the ways in which they interact with Systems of Systems (SoS), especially in the world of rapidly moving technology. Some of the challenges of working in this joined-up space are discussed, with particular reference to Enterprises of Development. The crucial importance for skilling staff in all enterprises to deal with this more complex world is a recurring theme.

1.0 INTRODUCTION

Enterprises take many forms and can interact in a many ways with Systems of Systems (SoS). In particular:

- *Enterprises, working individually or in groups, are required to build and operate SoS; and*
- *SoS can bring enterprises together to fulfil a variety of purposes, across the commercial, government and military spheres (including the building and operating of SoS).*

The implications of these two rather simple statements are far-reaching, and becoming more so as digital technology – the combination of world-wide networks, embedded computing, position location, distributed sensors and web based services – makes possible the bringing together of systems and enterprises in a greater number of ways to create new opportunities. These trends are, however, presenting commercial organisations with new challenges in the form of increased competition and higher uncertainty, to which they are responding by investing even further in digital systems. This is turn is adds to their uncertain futures which puts pressure on those who develop systems for them and the methods they employ.

Military organisations are experiencing the same opportunities and challenges, as digital technology, often driven by the commercial world, brings about a revolution military affairs. NATO has recognised this in its *Transformation* and *Network Enabled Capability* (NNEC) programmes, and many alliance nations have similar initiatives. At the same time, the world in which the military has to operate is becoming more and more uncertain: joint operations and wider alliances may have to be assembled at short notice; new and unexpected adversaries are emerging; and the systems we depend upon are potentially vulnerable to novel types of threat, including cyber-security.

This paper explores these themes in an open-ended way. The ideas presented are less mature than in other lectures – as reflects the general literature in the field – and are therefore more incomplete and speculative in nature. They are nonetheless put forward in the interests of describing the wider, more complex and rapidly-changing world in which SoS – and the discipline of building them – both have to operate in the 21st Century.

2.0 ENTERPRISES

2.1 Definitions and Types

For the purposes of this paper, we use the following, quite general definition of an enterprise [1]:

> *A complex, adaptive socio-technical system that comprises independent resources of people, processes, information and technology that must interact with each other and their environment [and other enterprises] in support of a common mission.*

Here we are treating organisations as systems, and making clear that people (and their social organisation and culture) are all included, just as much as the technology and process they use. The enterprises we are dealing with here all have as a defining feature some form of shared goal or mission. And the phrase added in parentheses shows that we will be concerned here with how enterprises interact with each other when they have a mission in common.

The following are all enterprises meeting the above definition:

- A multi-national company, e.g.: hotel chain, car manufacturer.
- A supply chain: all those working with/for a prime contractor to produce a product.
- Governments, e.g.: single departments, whole nations, multinational federations (EU).
- A specific programme – e.g. Apollo Moon Landing, London 2012 Olympics.
- A military operation, e.g.: single or joint service, allied.
- An alliance, e.g.: NATO, AU (African Union).
- A multinational agency, e.g.: United Nations, ICAO (Civil Aviation), FIFA (football).
- A trans-national initiative, e.g.: eradicating malaria, countering global warming.

The nature of enterprises is discussed from a systems point of view in the SEBoK [2] (*Systems Engineering Body of Knowledge /Body of Knowledge for Academic Curricula in Systems Engineering)* and further by Martin [3], from which we get Figure 1.

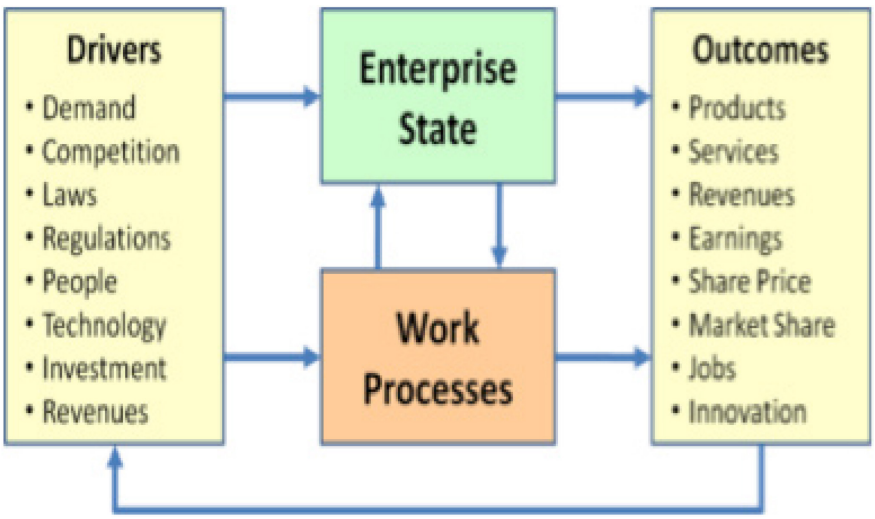

Figure 1: An Enterprise as a System.

In this figure, an Enterprise is shown as a system which translates *Drivers* into *Outcomes*, using *Work Processes.* The *Enterprise State* will usually be changed as a result of performing this translation, and may also be required to change itself in order to be capable of doing so.

Although the model was designed to depict commercial organisations, it can readily be applied to military ones.

2.2 Enterprises as Dynamic Systems

A feature of enterprises of prime interest here is their ability to collaborate dynamically, i.e. join and leave federations as the need arises. This can be done on widely differing timescales, from short to long-term, as shown in Figure 2. As we cross the spectrum from left to right:

- Two or more companies engaged in a simple, one-off exchange of information, for which the Internet will usually suffice, with all other company systems remaining unaltered.
- A collaborative project between a group of companies, which might require a bespoke or off-the-shelf SWE *(Shared Working Environment)* around the boundaries of the project, often usually defined by whichever organisation is in the lead.
- A major joint venture, for example lasting a number of years; here one would expect agreement to common work practices and long-term exchange of staff. Depending on the scope and duration, information systems support might take the form of an extended SWE or an EA *(Enterprise Architecture [4])* programme of the required scope.
- Company merger, where the whole of the organisations involved are joined, infrastructure (buildings, IT, etc.) rationalised and staff reorganised into new structural units. On the technical side one would expect a full-blown EA programme to deal with the new fully integrated IT.

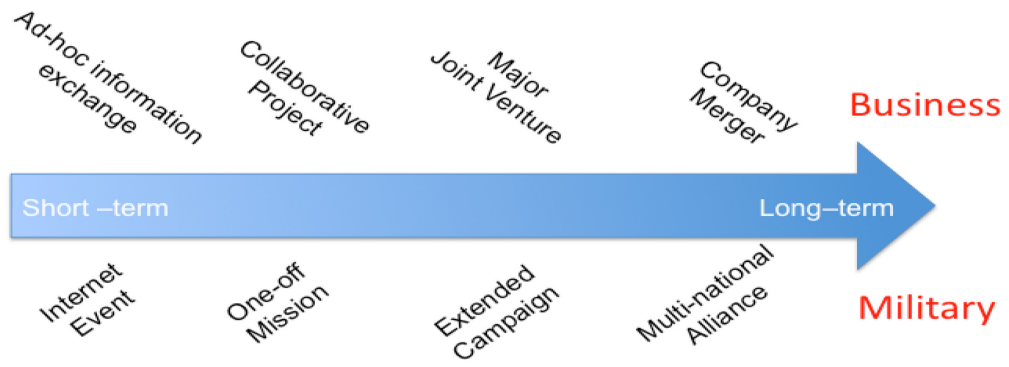

Figure 2: Dynamic Collaboration Between Enterprises.

Collaborations between military enterprises can be placed on the same spectrum, again depending on the duration of the joint mission and the depth of integration. For interactions of short timescale, we have simple information exchange (for example email, attachments) and at the other, multinational alliance operations, with much greater integration of command structures, sharing of military assets and full interoperability of their information systems.

As an example of an enterprise, NATO presents a mixed model: a number of centrally-funded infrastructure programmes form a long-term backbone, supplemented by specific projects to build deployable systems, relying on national systems to interoperate for full-blown alliance operations. This will be retuned to later.

2.3 How Enterprises Interact with Systems of Systems

A number of different enterprises can interact with a SoS, and do so in a variety of ways. Figure 3 shows in generic terms the main enterprises concerned:

- *The Enterprise of Development*, responsible for designing and building the components of the SoS;
- *The Enterprise of Support:* responsible for maintaining and reconfiguring the SoS once the parts are built;
- *The Enterprise of Control:* all those responsible for operating the system, i.e. changing its state in real time; and
- *The Enterprise of Use:* all those who use the SoS to fulfil the agreed mission (e.g. where military operations take place).

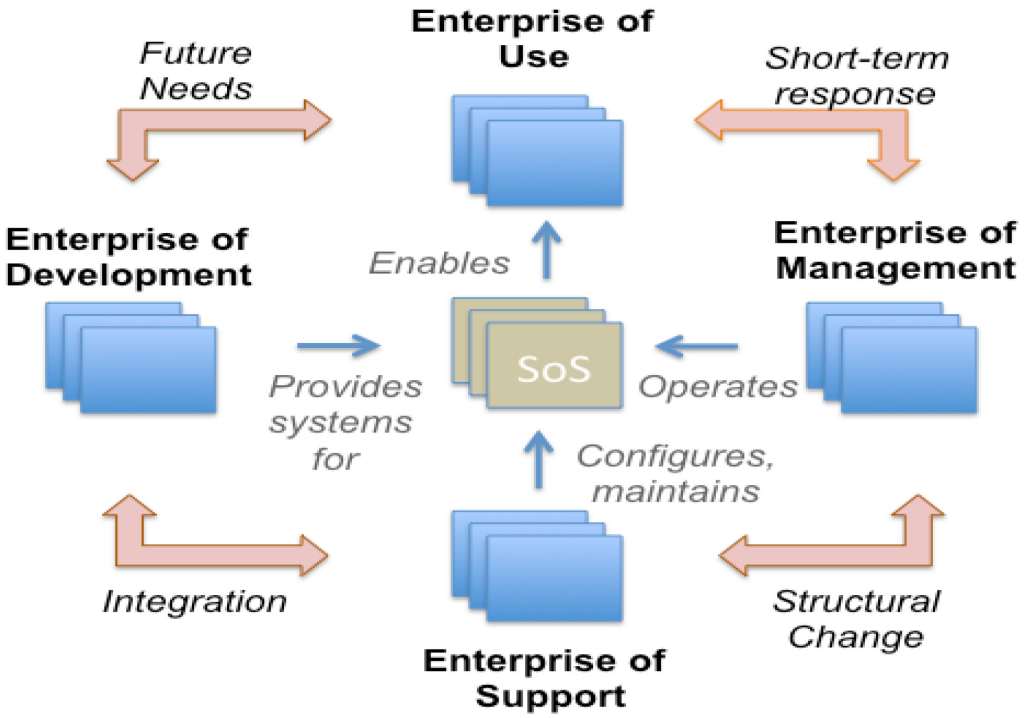

Figure 3: The Enterprises Which Interact with a System of Systems.

One rather fundamental point shown in Figure 3 should be singled out for emphasis:

> *The Enterprise of Development does not necessarily build a single system. It may of course do so, but in the general case, it builds the components from which those in the world of use – the Enterprises of Support and Management – work together to put together one or more SoSs to support the needs of the Enterprise of Use at any moment of time.*

There are a number of implications for the interactions between the enterprises concerned:

- The *Enterprise of Development* should build the elements of the SoS in a way which allows them to be put together in a number of different configurations by the *Enterprise of Support.* This is especially important if future operations are uncertain or if flexibility is required. (Ref [4] uses the term *Composability* to describe the property of a SoS of being able to be put together flexibly according to changing circumstances.)

- The *Enterprises of Development and Support* will usually cooperate over initial integration of the SoS and its introduction into service; the process may be repeated many time over the whole life of the SoS, as new systems are introduced, or the configuration changed.
- The *Enterprises of Support* and *Management* need to collaborate if the configuration of the SoS changes in such a way as to cause a change in how control should be undertaken.
- The *Enterprise of Management* is responsible for changing the state of the SoS, in response to changing short-term needs – changes in system loading or loss of service through damage – where this can be done with the same component systems.
- *The Enterprise of Use* liaises with the *Enterprise of Development* over future needs arising from likely future uses and technical opportunity; this usually gives rise to the development of new component systems, within an evolving architecture.

The world portrayed here appears quite complex, especially if the enterprises concerned are themselves compound, i.e. made up of more than one organisation. The apparent complexity may be even greater if the enterprises use some form of SoS to support their roles.

The sheer complexity of the future environment and number of possible combinations of enterprises and systems is such as to defeat any attempt to systemetise all the engineering and management processes concerned. Moreover, we fully expect new and different forms of social networks to emerge – enabled by technology – and to evolve freely as a result of human and social ingenuity to create novel enterprises of all types.

In this author's view we should aim to create conditions in which creativity, collaboration and collective action can flourish and take over where necessary from fully codified practices. The alternative is to risk stagnation through over-prescription.

To demonstrate the kinds of organisation and pragmatism which take place in the real world, two vignettes are set out below, taken from very different parts of the SoS domain. The lesson here is that success depends on a combination of built-in technical capability and human skill.

Vignette 1: Domestic Entertainment System.

A consumer has a continuing need for a home SoS to store and play TV programmes, video and photographs. He/she buys the parts (computer, flat-screen, camera, subscription to on-line broadcaster) over time from the local computer shop, department store and TV service provider, and installs them alongside part of his existing system. He receives some advice on-line from the computer supplier and a helpful person in the TV shop (together they form the *Enterprise of Support*). His needs keep changing as new technology appears, for example via new apps on his smartphone, and the children ask for what their friends are using.

He is quite computer literate and finds that he can quickly get to an initial capability. He is helped by the *composability* provided by *developers* of the parts who have agreed on common interface and information storage standards, though some suppliers still attempt to lock him into their own closed proprietary systems. Some example *architectural patterns* are available in the user manuals. The user leads his own *support*, continuing to use on-line, telephone and in-store advice where necessary. When visitors call in, everyone works together, extending mutual *support* to link their computer into the SoS to play family photos and YouTube videos. The consumer also acts as his own SoS *Manager*, through an assortment of remote controls and his PC. And he feeds back his future needs, complaints and suggestions through on-line review sites – to which the *developers* will respond in some general manner, taking account of all such comments and future markets.

Vignette 2: Coalition Command & Control System

A group of allied nations are required by their governments to put together a deployable force to support an urgent mission in an African nation. They have differing command and control systems as a result of separate national procurement programmes. However, participation in NATO standardisation activities means that national *developers* have built in a degree of built-in interoperability; variations nonetheless exist.

One of the nations is designed as lead, and brings together a group of international J6 staff to assemble and integrate a Joint HQ, using a version of their national Joint CIS as the framework. The staff benefit from having been recently involved in an allied interoperability demonstration, which means that some know each other and have tested configurations which can be re-used. But they still find difficulties arising from the systems offered, and specific requirements of this mission. *support* team solves the problems by doing some simple MODAF modelling (an example of architecting taking place outside the *development* environment) and building some ad-hoc gateways which work but with some reduction to full capability. Security is a thorny problem.

On completion of the initial mission, it is decided that a residual force will be left behind for humanitarian reasons. The *support* team dismantles the initial SoS and reconfigures it to work over local civil telecommunications and interoperate with both a UN force and *Médecins Sans Frontières.* On final completion an evaluation team is established to assess lessons learned and pass them back to all concerned, including those responsible for *development* of next generation systems in respective nations. As it happens, the opportunity to make changes to existing systems will not occur in some cases for over two years, due to the phasing of key development projects and other requirements seen to be more pressing by the *Enterprises of Users.*

2.4 NATO as an Enterprise

NATO is of course an enterprise in its own right. It is internally complex, comprising a number of nations of differing capability, sharing a common goal. The Alliance is called upon to fulfil missions with other alliances, and nations may enter into separate coalitions with each other for specific missions of varying duration. The whole is embedded in a network of other enterprises, for example: NGOs, industrial and national development organisations, academia and commercial standardisation bodies. And we should not overlook the enterprise of possible adversaries (threat), which is becoming more diverse, interconnected and capable, in many instances benefitting from the same commercial and IT service infrastructure (e.g. the Internet) as NATO.

Figure 4 [5] depicts layers of the NATO Enterprise Command and Information Systems (CIS), moving outwards from the core systems (those procured through Alliance funding for common use), through NATO Force Structure which includes deployable and rapid reaction forces to purely national and coalition systems, and beyond.

Core systems are supported by the NCIA (NATO Communications and Information Agency), leading the Enterprise of Development for Core Systems, offering support and assisting with wider interoperability between nations. The latter is achieved in part through implementation of the NATO Interoperability Standards and Profiles [6].

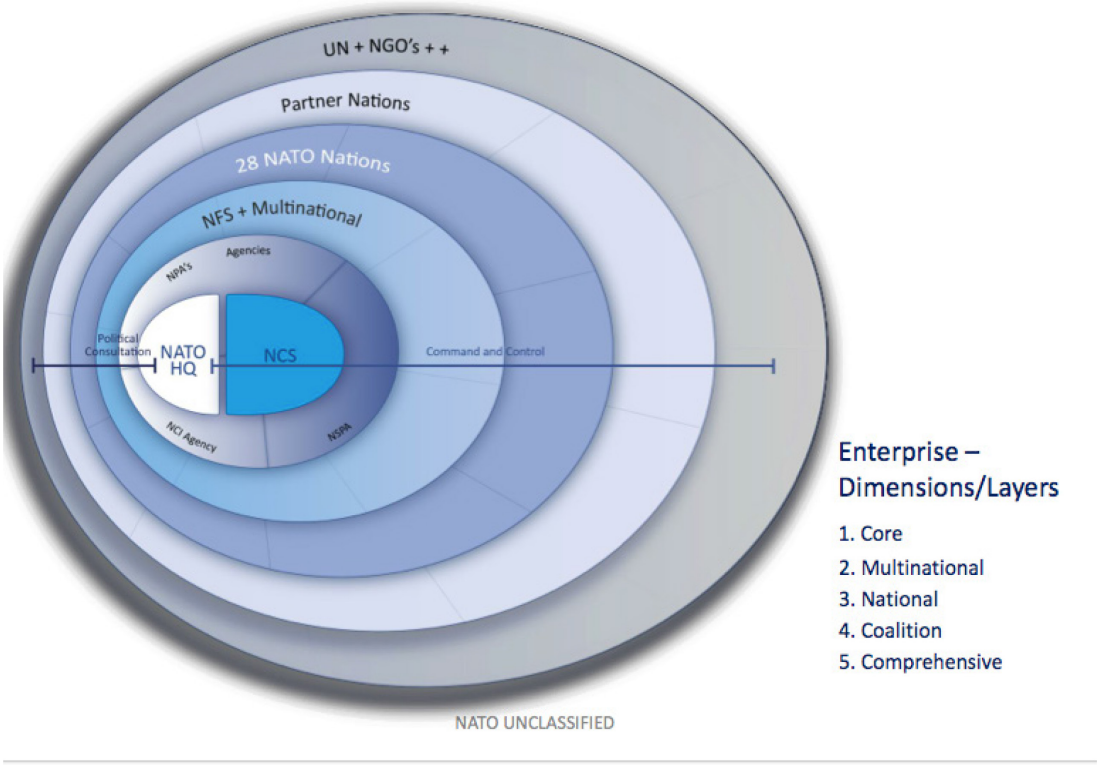

Figure 4: The Enterprise of NATO CIS.

3.0 ISSUES FOR THE ENTERPRISES OF DEVELOPMENT

3.1 Why does this Enterprise Matter?

Although all the types of enterprise discussed have a role to play in ensuring effective operation of a SoS, and need to act together collectively if success is to be achieved, one of the enterprises rightly receives greatest attention, namely the *Enterprise of Development.* The reasons for this are not hard to find: it makes the biggest contribution to building in the ability for systems to integrate down-stream, for example through the early application of architecting processes, and it is through the development process that new technology usually enters the SoS.

That is not to say that a great deal of know-how in making SoS really work does not lie in the other types of Enterprise, as was illustrated in the two vignettes; but what goes on here is much less well codified. A further, more pragmatic, reason for spending the rest of this paper on this topic is that this is where the author's experience lies, largely working on Information Systems in Defence and Security domains.

3.2 Why Do Enterprises of Development Find it Difficult?

Undertaking the early development stages necessary to successfully build SoS requires projects to join up and work together coherently. In addition, development organisations find themselves in a world of high uncertainty and rapidly moving technology.

There are a number of factors in play here [7]:

- *Technical:* projects proceeding at different timescales, often dictated by the technology involved or resource considerations, and poorly characterised systems;

- *Process:* projects being brought together from different enterprises applying their own in-house processes (or none); and
- *Management:* poor appreciation of the need to work across the organisation to create coherence, not helped if the technical community makes its solutions (e.g. Architecting) difficult to understand or underestimates the effort required to apply them in practice.

The combination of uncertainty and complexity defines a classically hard, or 'wicked', problem. A study of leading opinions across the project management community [8] identified three challenges which directly echo the points already made:

- *Persistent ambiguity and equivocality of project goals and contradictory and conflicting understanding of project success;*
- *Inherent unpredictability of future events, actions, responses and behaviours, time flux and change which expose the paradox of project control; and*
- *Complex multi-agency interfaces, social interaction, and processes of relating among project actors with different professional backgrounds.*

The language may be a little dense but the points are clear. What enterprises have found it useful to do are set out in the following sections.

3.3 Generic Enterprise Responses

Reference [7] discusses general strategies which allow us to both reduce the avoidable complexity arising of the interactions between projects and enterprises – essentially a technical management challenge – and cope with that which remain. The same strategies help us to deal with the difficulties which arise from uncertainties in the operating environment of the SoS. Collective experience of those working in the field suggests the following responses – which are closely linked – can all contribute. Each enterprise needs to understand which of these works best for its particular circumstances and apply them appropriately.

Focus on Goals

In framing objectives for SoS development it is better to aim for outcomes, rather than rigid requirements based on a perception of possible future worlds. It is essential to retain flexibility in the face of change, and employ more adaptive programme and project management practices with associated contracting models.

Break Projects and Programmes into Manageable Time Segments

Alongside the last strategy it is useful to break groups of projects into groups which can be implemented in short, manageable stages. This is the alternative to 'big bang' approaches. The advantage comes in being able to adapt to changing circumstances, insert technology progressively, and get feedback from the experience of early fielding. Some of the systems engineering models described in other papers show how this can be done on a continuous basis. The Trapeze and Wave models [9] are based on this principle.

Aim for Whole Capability

Users are in general more interested in having improved capability than just new equipment. Achieving this requires those engineering the SoS not only to build new systems – and integrate or modify legacy systems – but to take into account non-equipment enablers such as *training, logistics* and *test equipment,* and ensure they are mutually consistent and integrated into the whole capability. (These enabling systems – for example test equipment, might also form a SoS in their own right, requiring integration with each other and into the overall capability.)

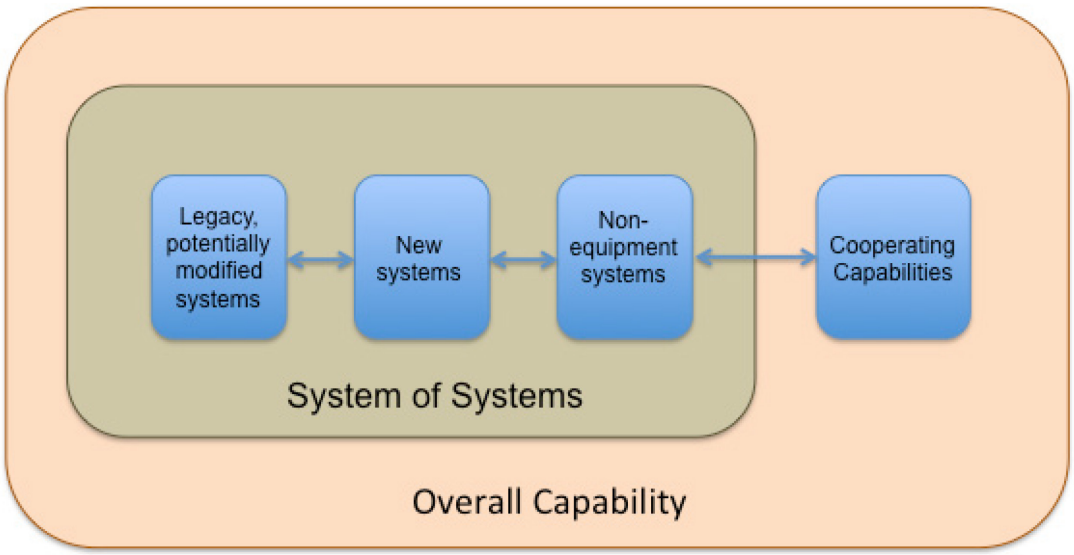

Figure 5: System of Systems Contribute to Overall Capability.

Build Adaptability into the Product

It is a sound engineering principle to make systems and products which are easy to adapt – at design, development, integration and operational stages, which requires applying the principle of *modularity*. This not only allows developed components to be updated once built, but in the case of SoS makes multi-project design more feasible by reducing the extent to which design changes in one system propagate throughout others, thus limiting the intrusion of changes and the risk of 'project churn'.

The achievement of modularity depends classically on producing product designs which maximize cohesion, while reducing coupling. This is equivalent to aligning the physical architecture of the constituent systems to major system functions (i.e. the functional architecture).

Cluster Projects Using Systems Principles

Further complexity reduction results if project teams in the development organization are clustered into domains, and managed as coupled programmes, the need for which is highlighted in TOGAF, in addressing the same issue for Enterprise Architectures:

> *As it is not possible to develop a single architecture that addresses all the needs of all the stakeholders, the enterprise must be partitioned into different areas, each of which can be supported by architectures. Enterprise architectures are typically partitioned according to Subject Matter, Time Period and Level of Detail.*

This leads to the desire for what might be called 'organisational modularity', which comes from aligning the programme management groups (and architectural domains) along functional lines. It is hard to prove why this is might be true, but intuition suggests that in many businesses, and certainly the military world, the main functions or capabilities are relatively more enduring than specific technological solutions, and therefore form the least unstable basis for managing complexity.

It cannot be emphasized too strongly that unless the programme/project organization is designed logically – and aligned with architectural domains – then large-scale complexity reduction is unlikely to be achieved. Within such structures it is possible to establish internal communication channels which are stronger and more frequent than between clusters.

This is not just theory. There is strong evidence from the industrial world that aligning the organisation with the product or service structure encourages constructive communication [10], leading to greater success in environments in which complex systems are developed.

Undertake Architecting in Layers

Inherent in the concept of clustering it the idea of layered architecture (and architecting), which will typically follow the 3-layer model show in Figure 6, with architectural activities as follows:

- **Enterprise Level Architecture** at the highest level, setting out technical policy and organisation-wide standards and, crucially, setting out the landscape by which system solutions can be segmented into domains. It will also develop (or participate in the development of) overall roadmaps for common infrastructure and services, updated as necessary to reflect changes in policy, funding priorities and technological developments.

  Enterprise architecture is also required to manage the integration points which cross domains and transition points between them, working with domain architects. In support of these roles, the Enterprise Architecture team maintains a minimum number of architectural views and plans, which may be aligned to the organisation's *Portfolio Management* system.

- **Domain Level Architectures** provide the essential middle layer in this model, sitting between the enterprise and individual projects. Architecture here is concerned with transforming the systems solutions within the domains. As shown below, within domains, it is possible to apply techniques of Programme Management (see Section 3.4). Domains are responsible for setting local standards and policies, working with projects in their early stages to develop designs for constituent systems, and overseeing domain-level integration technical and customer acceptance testing. One would also expect to see rich architectural views developed at this level and used as an integral part of decision-making.
- **Project/ Individual System Level Architectures** are concerned with detailed design of specific, time-related solutions to fixed requirements, and agreed integration points, determined by the domain architecture.

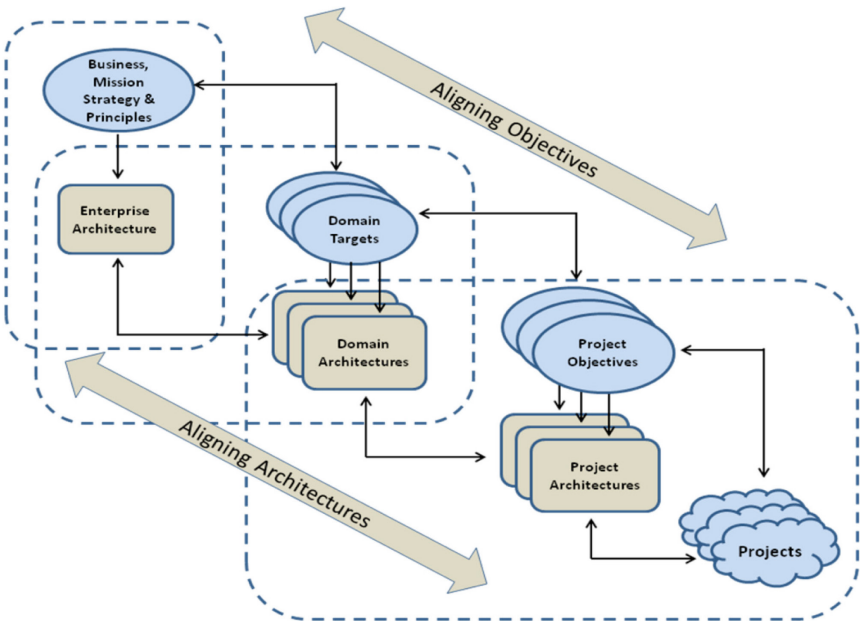

Figure 6: A 3-Layered Approach to Architecting and Management.

Maintain Good Inter-Project Communication

The need for good communicate has already been referred to. This is especially important and effective in the early formative stages of projects, when costs of making changes are small. Unfortunately, engineers are not naturally conditioned to exchange incomplete information, and project managers can be uncomfortable in case preliminary designs are confused with firm commitments. It is also important that major design changes or development difficulties in one segment are communicated instantly to others.

Making this happen requires training staff in softer skills, such as negotiation and persuasion, as well as cooperative behavior, balanced more formal set-piece meetings such as design reviews. When all this works well, it forms the essential glue holding together SoS developments, applied alongside more formal governance.

Align Development Approaches

Notwithstanding the last point, stage-based reviews remain essential, to allow goals to be revisited and updated formally to reflect changes both within and outside the project. Creating the balance might be called 'requisite formality'. One way of achieving this is to impose a small number of key review points which are generic in nature, externally visible and compulsory, while allowing programmes and projects to choose their own detailed life cycle approach, suitable for the complexity and product type, e.g. software intensive systems. (Giving staff such freedoms of course requires a more sophisticated workforce, capable of making intelligent choices, rather than following 'one size fits all' processes.)

One might call his principle 'life cycle modularity', in which external formal interactions are minimized and controlled, allowing freedom to pursue individually-tailored approaches within projects. The need to allow some flexibility in the choice of life cycles can be especially significant in the case of portfolios of systems with a rich mixture of (hard) platform and (soft) information processing systems, with radically different characteristic timescales. Here we might aim for highly-formalized systems engineering cycles for the platform development and more agile, rapid application techniques for the embedded command systems.

Use Flexible Design Strategies

Complexity reduction also depends on the way design choices are made. There are competing forces at play here. Leaving design choices open for as long as possible maximizes the opportunity for coordination, and in cases where technology is moving very fast making final decisions as late as possible (just in time design) allows for latest developments to be adopted. However, complexity reduction also argues for some key decisions to be made early to cut down the number of variables. Only experience allows this balance to be struck.

3.4 Integrating Processes

The process landscape surrounding the engineering management of SoS is still disparate and evolving, with examples of good practice emerging, for example from the DoD and EU's DANSE Programme [11].

One very significant issue is the need to integrate technical (systems) engineering and architecting with project management and associated processes, such as investment management. In large organisations, which are suited to the 3-level architecture approach described above, an integrated approach to management can be based on the three corresponding levels of Portfolio, Programme and Project. Within the UK, standard guidelines for managers working at these levels are provided by the MPA (*Major Projects Authority)*, and are backed by training courses and review material.

This is by no means the only way to proceed (others are described in Ref [12]), but it has been demonstrated in one large organisation, by producing an integrated and mutually-consistent set of guidance and training material, which brings together:

- Enterprise Architecture and Portfolio Management [12] at the topmost level;
- Programme Management (based on MPA's *Managing Successful Programmes [13]* – MSP) at the middle level; and
- PRINCE™ with TSE (*Traditional Systems Engineering*), and/or UP (*Unified Process*) at the project level.

Figure 7 shows the layered processes involved. Although it appears complex, it has some attractive features which support the general strategies already discussed:

- The functions of the topmost Enterprise Architecture function naturally fit with portfolio management; each has a view and an influence on organisation-wide strategy, risk, commitment and resources – and they naturally share much of the same information.
- The MSP philosophy is outcome-based and inherently goal-oriented, and encourages splitting long developments into tranches, thus allowing phased implementation, with re-orientation between them to allow for changing circumstances.
- MSP also fits well with the TOGAF ADM (shown above as developing, managing and maintaining the Programme Architecture). Each standard encourages the concept of transitioning from legacy through projects to the service domain.
- PRINCE encourages project managing to requirements, and provides flexibility in the choice of technical development approach; a simple, generic systems engineering process fits very well here.

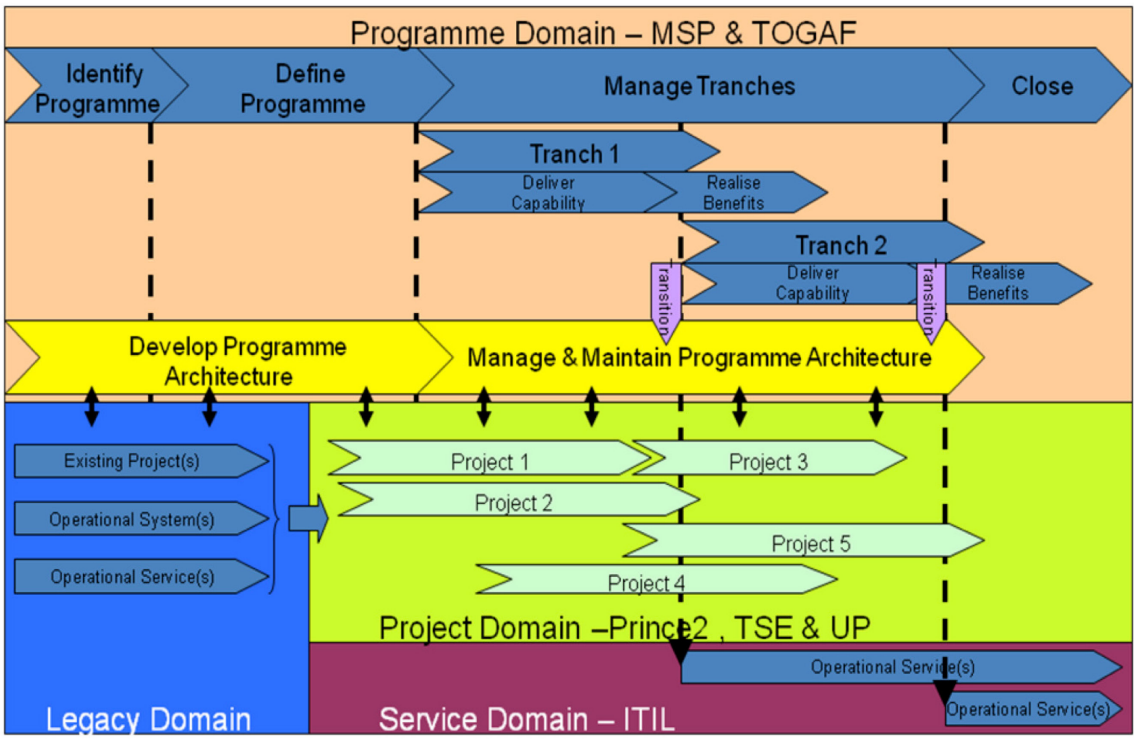

Figure 7: Integrated Processes at Programme/Domain and Project Levels.

Perhaps most importantly, the use of accepted management processes allows large areas of existing good practice to be integrated with enterprise systems engineering, and encourages dialogue with a key community with which systems engineers need to work most closely to achieve success.

3.5 The People Dimension

In keeping with the idea of an enterprise as a socio-technical system, effective development of SoS requires skilled people, properly supported by tools, training and good guidance, working together in a collaborative manner. This much is common with conventional systems engineering, but the move towards SoS engineering requires a more sophisticated workforce and more integrated enablers.

Figure 8 shows the capability necessary to undertaking SoS engineering (*Architecting* in the terminology of TOGAF, from which it is taken). The RHS represents the world of managing projects and portfolios, delivering new solutions (i.e. *SoS capabilities*) to Business Operations (i.e. *Enterprise of Use*). Of special interest are the human resources on the LHS, with an associated knowledge base and professional development delivering staff to undertake roles in development. The model is integrated with governance bodies above and supported below by an information base (*Architecture Repository*) and *Enterprise Continuum* (frameworks, patterns and standards).

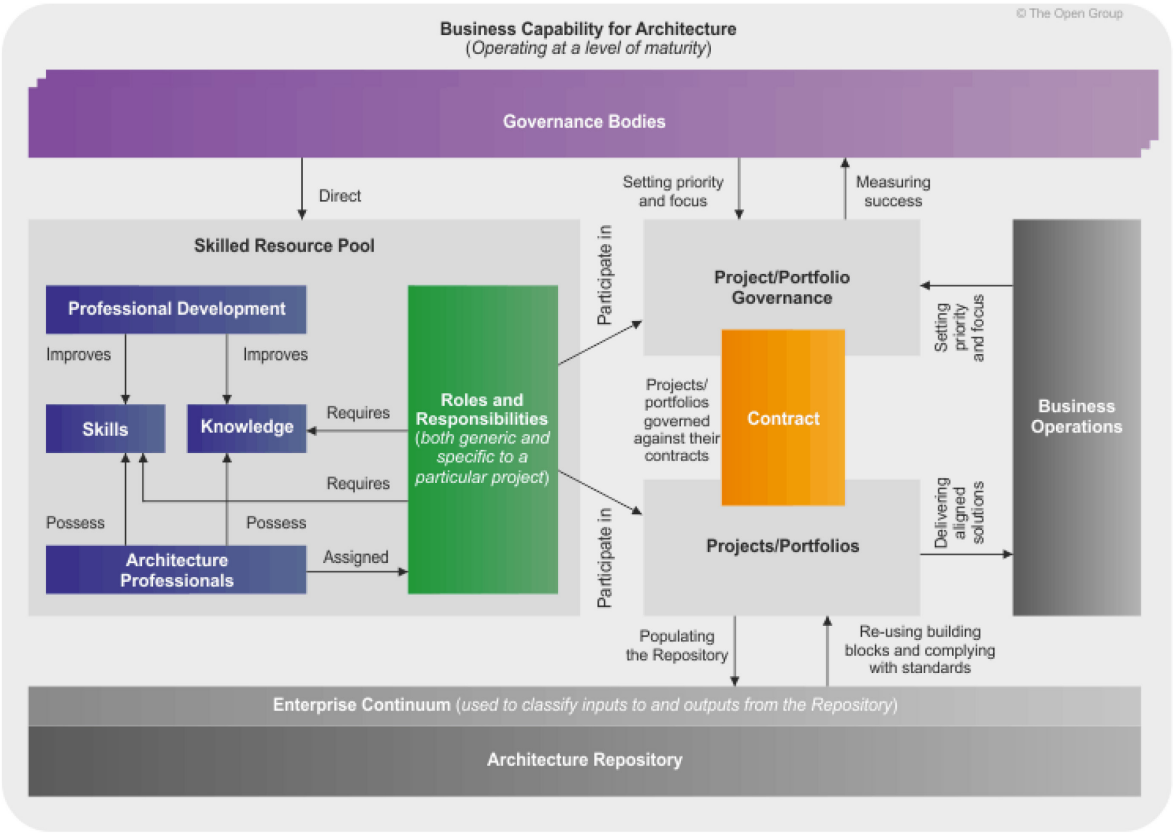

Figure 8: Business Capability for SoS Engineering (from TOGAF).

Lessons learned from the implementation of such a model include:

- *Invest in people:* skilled staff are ultimately more effective than highly elaborated processes, so investment in people (training, guidance, mentoring) is essential. It is not possible to cover all cases in a single, generic process set, and it may well be necessary to adapt an organisation's processes to others' in an extended enterprise. Smart people can build on a repertoire of logical models and take the necessary steps to the practical.
- *Listen to what works:* it is easy to implement communication strategies which are essentially one-way, without considering the need for feedback. Centrally-developed process documents and

training material are only as good as the experience of those who write them. Over time there may well be a greater collective experience (and wisdom) in the practitioner community, and this needs to be incorporated into the knowledge base.

- *Adapt skills to tasks:* working in the upper reaches of enterprise and programme management requires highly-skilled and experienced individuals capable of making key judgements in circumstances of high uncertainty and complexity. These have to be selected and career-managed accordingly. However, not all staff need to attain these levels, and there is still a need for a larger cadre of staff capable of undertaking implementation tasks in the greater certainty of constituent projects.
- *Invest in the soft skills of (technical) staff:* the world of enterprise systems engineering places a greater dependence on the ability to negotiate, present and discuss issues and make compromises, often in group situations. Experience shows that these softer skills are as important as more traditional technical skills and that staff should be trained and selected on this basis for the more sensitive roles.

4.0 SUMMARY AND CONCLUSIONS

For SoSs to be successful, a number of different sorts of enterprises need to cooperate in their development and to support their operations. This is becoming especially complex as digital technology drives the increasing 'connectedness' of systems and projects, and the world in which they will operate becomes more uncertain.

These trends present a number of challenges to those who have previously only dealt with projects and systems in isolation, not least across the *Enterprises of Development.* Responses are emerging from a number of such organisations, including those represented in this lecture series. These require special attention to undertaking systems engineering and engineering management in an integrated manner, wherever possible linked to organisational strategy and the needs of the *Enterprises of Use*.

However this is applied, there is a strong and increasing need to train and support a new cadre of engineers and managers who can deal equally well with the softer, interpersonal issues and the more complex engineering aspects. The professional societies, working with the education and training communities, all have a role to play in building the necessary knowledge base and delivering to those at the workface – acting as an *Extended Enterprise for SoS Engineering.*

5.0 REFERENCES

[1] R E Giachetti, *Design of Enterprise Systems*, CRC Press, 2010.

[2] www.bkcase.org.

[3] J Martin, *Transforming the Organisation Using a Systems Approach,* INCOSE International Symposium, Denver, June 2010.

[4] This and other terms relating to architecture are explained more fully in the paper in this lecture series: P Brook, *Role of Architecting in Systems of Systems.*

[5] From: K Gijsbers (M Gen), *Realising the Connecting Forces Initiative*, NATO NEC Conference, Lisbon, April 2013 [UNCLASSIFIED].

[6] https://nhqc3s.hq.nato.int/Apps/Architecture/NISP/, dated August 2014.

[7] P Brook and T Riley, *Enterprise Systems Engineering – Practical Challenges and Emerging Solutions*, INCOSE International Symposium, Rome, July 2012.

[8] S Cicmil et al., *Exploring the Complexity of Projects – Implications of Complexity Theory for Project Management Practice,* Project Management International Inc., 2009.

[9] J Dahmann et al., *An implementer's view of Systems Engineering for Systems of Systems,* IEEE Systems Conference (SySCon), Montreal, April 2011.

[10] M E Sosa et al., *Are your engineers talking together as they should?* Harvard Business Review, Nov 2007.

[11] Discussed in the paper: J Dahmann, *System of Systems Lifecycles,* in this volume.

[12] Portfolio management is the corporate, strategic-level process for coordinating successful delivery across the organisation's entire set of programmes and projects. The 'portfolio view' represents the complete picture of the enterprises commitment of programme and project resources to deliver its strategic objectives.

[13] Available from www.tso.co.uk and other sources.