"Composable Capability" – Principles, Strategies and Methods for Capability Systems Engineering

H G Sillitto
Thales
1 Linthouse Road
Glasgow G51 4BZ
UNITED KINGDOM

hillary.sillitto@thalesgroup.com

ABSTRACT

Current systems engineering approaches for capability are difficult to understand and not deployed consistently. Language is a barrier to understanding. A top-down approach to "capability" still seems intractable. Mission thread approaches are not sufficiently espoused because of the apparently infinite number of potential mission threads, most of which will never be encountered in real life. This paper offers an alternative way of looking at the problem that contains the complexity explosion and has the potential to give more benefit with less difficulty. The proposed approach is to treat the force element as a system, and use a carefully chosen representative set of mission threads to derive the functional, behavioural and performance requirements for force element interoperability.

1.0 INTRODUCTION

The traditional system focus of defence acquisition has been the "equipment". Force elements have been constructed bottom up from equipment – inevitably such bottom-up constructs have unintended emergent properties and are not optimised as "force elements".

It is possible to identify stable, well characterised building blocks (Force Elements or FE) from which a wide variety of military task force structures can be put together providing almost infinite variety of capability solutions. This approach mirrors how Defence constructs task forces from available units, and provides the flexibility sought by commanders to deal with the unexpected and unforeseen. This paper proposes that the "system focus" in defence acquisition should be the force element, not the individual equipment. Equipment requirements and acceptance criteria can then be derived from the capabilities and interoperability characteristics required of the force element.

1.1 The military need for flexibility

> *"On every occasion that I have been sent to achieve some military objective in order to serve a political purpose, I, and those with me, have had to change our method and re-organise in order to succeed. Until this was done we could not use our force effectively. On the basis of my lengthy experience, I have come to consider this as normal - a necessary part of every operation. And after forty years of service, and particularly the last twelve, I believe I have gained an understanding of how to think about this inevitable and crucial phenomenon of conflict and warfare. The need to adapt is driven by the decisions of the opponent, the choice of objectives, the way or method force is applied, and the forces and recourses available, particularly when operating with allies. All of this demands an understanding of the political context of the operation. Only when adaptation and context are complete can force be applied with utility."(General Sir Rupert Smith, 2005)[1]*

1.2 The value of System Architecture

The purpose of "systems architecture" is to ensure that the various parts of our "system of systems", when connected to each other and placed in their operating environment:

- fit together
- work together
- achieve the required effect
- do not produce unacceptable side-effects

and can be

- kept operational over time
- reconfigured to meet "reasonable unforeseen" circumstances.

Why this matters is made clear by the preceding quotation.

1.3 Things are changing: Open Systems

Thales's experience in supplying the GVA (Generic Vehicle Architecture) for the Foxhound LPPV (Light Protected Patrol Vehicle) is very positive. The open systems approach will reduce the cost and time of adapting to emerging threats and mission needs. The GVA approach does not give complete flexibility to do absolutely anything. But if we accept the constraints imposed by the GVA standard, it allows us to do an awful lot of things very quickly and at low risk.

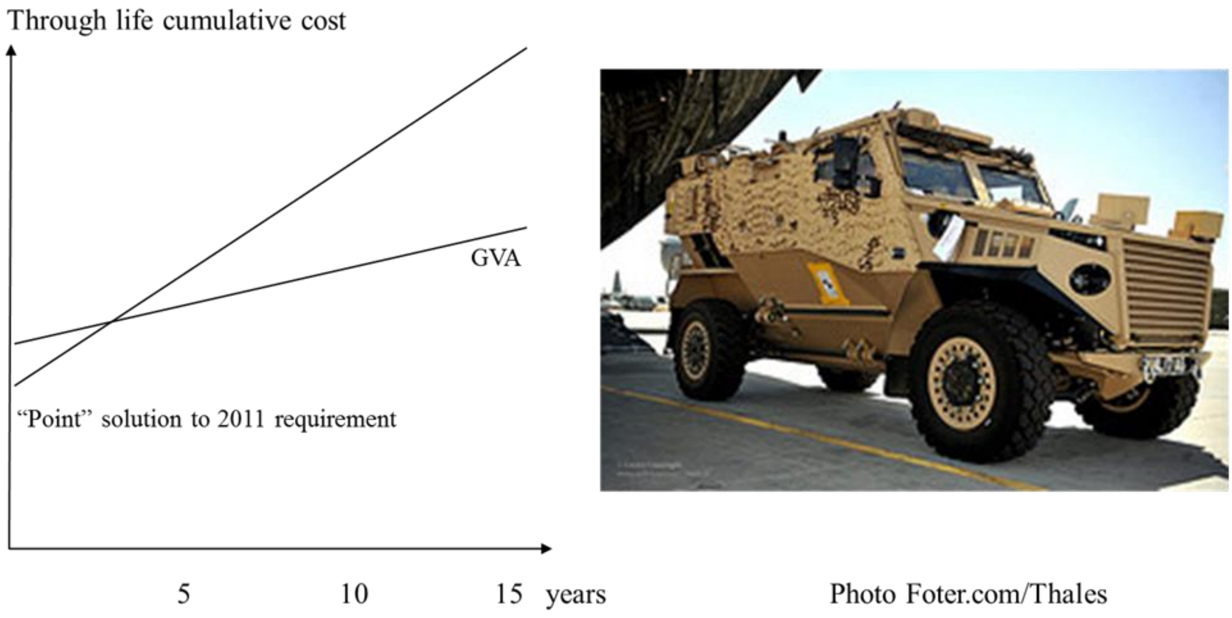


Figure 1: Foxhound LPPV, first production vehicle fitted with Generic Vehicle Architecture (GVA)

2.0 WHERE HAVE WE COME FROM? TECHNICAL AND PROCEDURAL INTEROPERABILITY

"Interoperability"[1] is *"The ability of systems, units or forces to provide services to, and accept services from, other systems, units or forces, and to use the services so exchanged to enable them to operate effectively together"* [2]. It creates additional capability: the ability to share information and synchronise actions across a networked force, to develop and sustain a tempo and precision that give an overwhelming battle-winning advantage.

Interoperability allows systems, operators and users to share information and synchronise actions to useful purpose. We did this rather well in 1944-45 [3]; tank columns advancing through Normandy could get accurate and effective air support within a few minutes using voice radios, smoke flares and well-defined, straightforward procedures.

With the advent of digital communications and information systems on the battlefield, the focus moved from procedural approaches for inter-unit interoperability to technology approaches to inter-equipment interoperability. When I took over The Integration Authority from Peter Brook in 2005, we faced the "challenging" task of getting to grips with interactions between any or all of 200+ concurrent equipment procurements, without full access to information on in-service legacy equipments.

Reflecting on this experience leads to consideration of a different way of thinking about specifying defence equipment – focusing on "purpose", i.e. what constitutes mission success, and viewing the "Force Element" rather than the "equipment" as the "system of interest."

3.0 PROPOSED APPROACH TO COMPOSABLE CAPABILITY

GVA uses a well known principle for the engineering of complex software systems: loosely coupled **objects** exchanging well-defined **services** for various **purposes**. This description is remarkably similar to the definition of interoperability given above. Since systems engineering can be applied to any "system" at any level, and to socio technical as well as purely technical systems, it is worth seeing if a similar approach would work in the larger system that is UK Defence – for example what happens if we treat Force Elements as systems, and use systems engineering to specify and design the force elements rather than just the equipment?

3.1 Why systems engineering

The basic argument for using systems engineering is that the cost of change is low early in the lifecycle and high later on, as shown by the following data based on statistics from a large number of projects. [4]

Table 1: relative cost of change by lifecycle phase [4]

Phase of project	Relative cost of change
Requirements	x1
Design	x5
Build	x12
Test	x40
Operations	x250

[1] 2006 Australian Defence Capability Development Manual. This definition comes from the superseded version of UK Defence Doctrine published in the late 1990s, and remains the best for our purpose.

It is possible to architect to reduce the cost of "the sort of changes we are likely to need but can't predict with any certainty", as we have seen in the GVA example above. This can be done by good modular design, good choice of interface points, good choice of interface standards, and good choice of "chunk size" or system granularity.

3.2 Mission threads

Mission threads are a key part of design methodology for system of systems integration, because they clarify the purpose(s) for which interoperability is required and the different systems, functions and processes involved. It is not necessary to define all possible mission threads. It is necessary to define enough important mission threads, across the full range of potential operational missions, scenarios and force structures, to determine:

- required information exchange capabilities for each system of interest
- critical timeline and accuracy requirements for each system of interest,
- cumulative end-to-end error budgets for the critical mission threads, and how these error budgets are apportioned down to individual systems
- operational rules to resolve resource contention when multiple mission threads are running simultaneously in the System of Systems (SoS)
- security boundaries and safety requirements that constrain allowable SoS configurations.

If we specify and characterise force elements in a consistent way, and maintain configuration control in service, other mission threads of interest can be readily constructed, analysed and optimised for specific operational circumstances.

3.3 Two kinds of "capability"

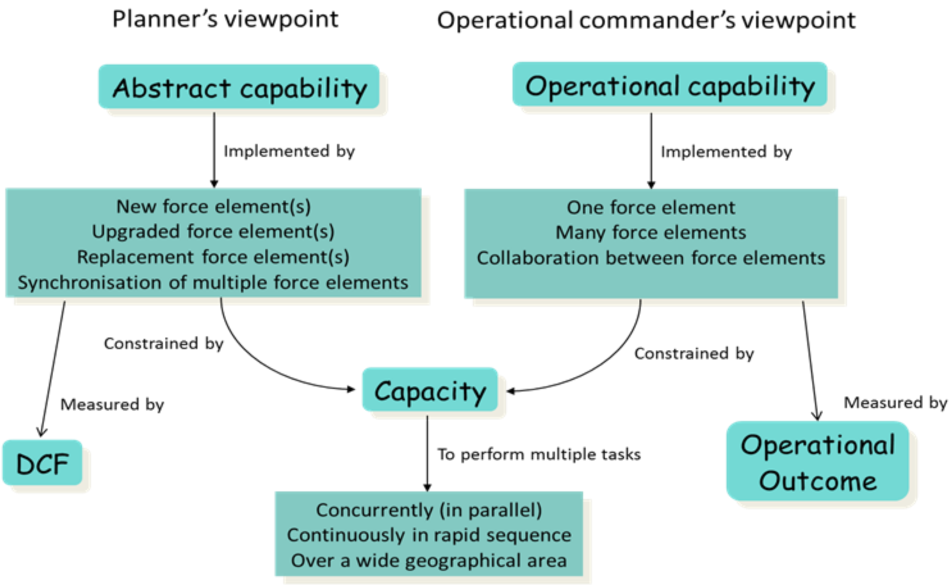


Figure 2: abstract and operational capability are different concepts

Abstract capabilities are the framework for definition of future defence needs and solutions, in future potential conflicts, and may be realised in different ways:

- creating a new kind of force element
- upgrading or replacing an existing type of force element
- deploy an upgrade across many or all existing force elements (defensive aids and communication capabilities often come in to this category)
- as an emergent property of force structure by synchronising the efforts of multiple force elements.

Mission thread analysis gives an understanding of how different (including legacy, modified and new) force elements need to work together to create the desired capability.

Operational capability is what can be achieved by existing force elements and force structure. As before, capability could be a property

- of one force element,
- of many force elements,
- or of collaboration between force elements

We also need to consider **capacity:** defined in terms of the ability of the force, force element or system of interest to handle multiple tasks or mission threads in parallel, and/or continuously in rapid sequence, and/or over a wide geographical area.

3.4 Elements of "the systems approach"

System concepts and paradigms are scalable and re-usable at different levels and can be applied to any type of system – technical, process, organisational, societal - -. This recursive nature gives systems engineering techniques both huge power, and huge potential for ambiguity and confusion. So it is really important to set the context for the use of such systems engineering models and to ground the abstract concept in tangible practical examples to "get everyone on the same page".

Mission threads are managed by defining standard "use cases" with associated standard parameter sets for each force element. Mission threads can then be constructed, analysed and optimised by seeing whether and how the relevant use cases fit together. A high degree of consistency is needed for this approach to work, and it needs to be based on "as operated", not "as specified" system descriptions. In mission thread analysis we need to understand how the individual force element behaves, and whether and how well it can manage concurrency between the demands to service the mission thread and conduct its own independent mission.

"Rules" are required to define how a system or force element should prioritise multiple tasking demands. This is one of the three key aspects of "behaviour" in the generic system model.

System of interest and system boundary: Each SoS mission thread is liable to involve a different subset of the SoS. This subset is the "system of interest" for the purpose of that particular mission thread. Further, the required effect may be achievable in a number of different ways, using different subsets of the SoS.

Generic System Reference Model: Hitchins [5] advocates the use of a Generic System Reference Model to provide standard templates for initialising the design and standardising descriptions of classes of system.

4.0 A SYSTEM FRAMEWORK FOR CAPABILITY

The composable capability framework is based on a generic system reference model that allows us to describe the properties of a "system" equally well at the three levels of task force, force element and capability component.

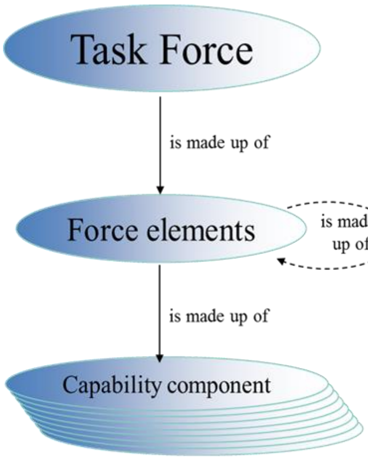


Figure 3: Framework for capability

A task force is made up of force elements. A force element may be made up of lower level force elements. The lowest level of force element is made up of capability components.

Capability components are defined in terms of the "defence lines of development" (DLOD)

- training
- equipment
- people
- information
- doctrine
- organisation
- infrastructure
- logistics.

System functions are defined at all levels in terms of the seven verbs of the defence conceptual framework:

- prepare
- project
- inform
- command
- operate
- protect
- sustain

4.1 A generic model for Force, Force Element and DLODs

This standard pattern is equally valid at force, force element and "component of capability" level - a repeating pattern that we can apply recursively at different levels. (It is also equally useful at lower levels such as subsystems within the capability components.) The table illustrates this, and proposes terminology for specialising the model to each level.

Table 2: How the generic system reference model applies to different levels of the framework for capability

Level	Structure	Behaviour	Function	Performance
Generic	Boundary Parts Relationships	Stimulus/response State Rules	Sense, Control, Operate, Protect, Sustain	
Task force	Scope; Force elements; Command relationships (belongs to / assigned to / commands / supports / supported by)	Command and communication protocols; Operational State; Operational rules including rules of engagement	Capability; Command (to orchestrate operational services between Force Elements)	Measures of Effectiveness
Force Element	Scope DLOD elements Command relationships Systems interfaces Programme dependencies	Command and communication protocols; Operational State; Operational rules including rules of engagement	Operational services (offered/ required/ exploited); Control (to synchronise technical functions)	Measures of Operational Performance
Component of Capability (system or subsystem)	Scope People, process and product elements; Interfaces and interactions	Command and communication protocols; System State; System rules	Technical functions Manage resources	Technical Performance Measures

4.2 Function

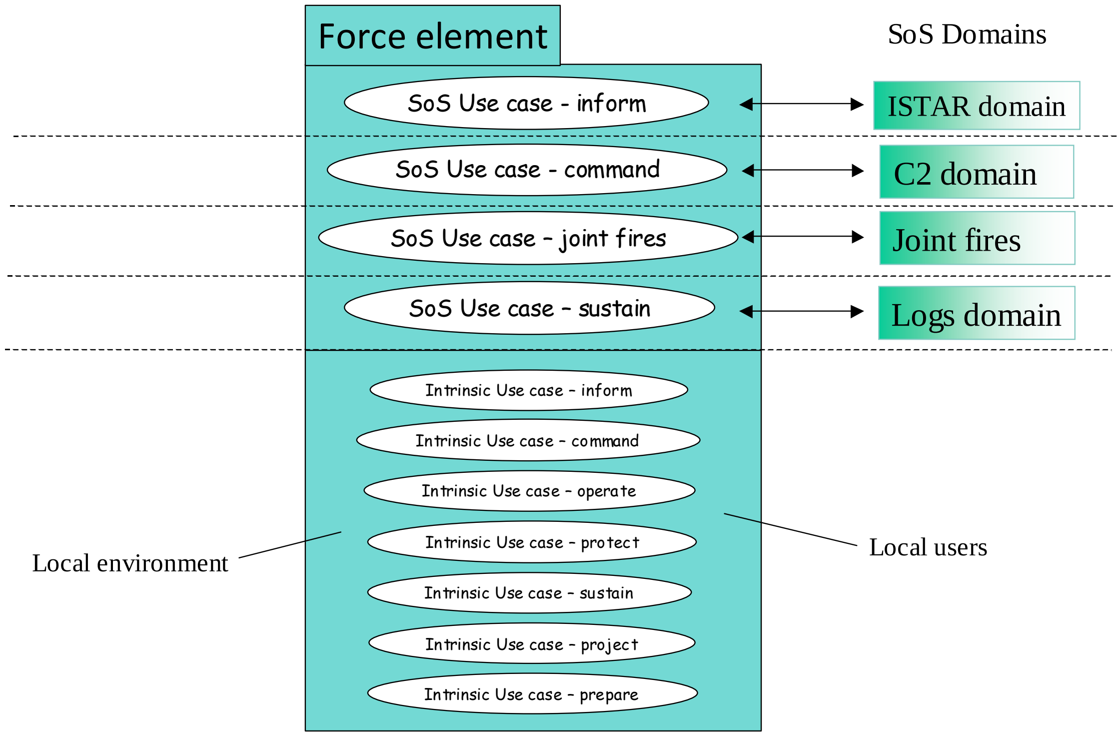
The core functions of a generic system include "operate", "survive" and "sustain", which are defined in terms of interactions with "operational", "threat" and "resource" environments respectively. (If it can't do these things it is not really a system [5].) The system must also include a control or management function, and needs to include a "sense" function if it is to respond to external stimuli from either or both of the environment and a higher command/control function. We can construct an equivalence table between the "minimum" generic model for a system and the seven functions of the Defence Capability Framework.

Table 3: How generic system functions map to the Defence Conceptual Framework (DCF)

Generic function	DCF function	State
	Prepare	Non-operational
	Project	Transition
Sense	Inform	Operational
Manage	Command	
Operate	Operate	
Survive	Protect	
Sustain	Sustain	

Within a force element we find:

- **intrinsic functions**: functions that are core to the purpose and correct functioning of the force element as an independent unit;
- **system of systems (SoS) functions**: functions that are orchestrated by C2 to create capabilities of the whole force that force elements cannot realise on their own: these may be related to joint effects, C2, ISTAR and Logs.



4.3 Performance

Performance parameters apply to behaviour – how quickly, how often, how many in parallel – and to function – how well, how much, how far, what probability of success.

A hierarchy of

- MoEs (Measures of Effectiveness)
- MoPs (Measures of Performance)
- TPMs (Technical Performance Measures)

can be aligned directly to the hierarchy of task force, force element and capability component.

5.0 SUMMARY - COMPOSABLE CAPABILITY

Composability requires that we characterise SoS functions in a consistent way in terms of:

- basic behaviours,

- key parameters,
- timing and accuracy,
- concurrency.

We can define a standard template for a force element, in particular its system of systems functions. This allows any force element to be specified and measured in terms of its characteristics that affect SoS capabilities, and allows any mission thread requiring interaction between force elements to be analysed, validated and optimised. This allows doctrine for new capability configurations to be quickly established; and allows the effectiveness of potential new or improved force elements to be evaluated against capability targets. Proper accounting for concurrency at system and force element level allows proper accounting for capacity at force level.

Acceptance criteria should include proper verification and validation of system of systems functions. Validated models of system of systems functions should be supplied along with each major deliverable and in particular should be updated at the point of acceptance to reflect the actual in-service build standard and any subsequent in-service changes.

A modular force structure will allow commanders to generate new combinations of capability at short notice using existing and proven modular force elements.

Paradoxically, in this new world, the more consistent and well-defined we can make the properties of the individual force elements, the more freedom that will give operational commanders to adapt the capability of the task force to the task in hand – not by reconfiguring the force elements themselves, but by adjusting the way they interact.

This ability, however, depends on using the right level of granularity or "chunking" of the force elements, and on knowing how to understand, measure and adjust the effects achieved by the interactions between force elements.

Note: a complementary paper which expands on and provides more background to the material presented here will be presented at the INCOSE 2013 International Symposium [6].

THE AUTHOR

Hillary Sillitto was until recently Thales UK's Systems Engineering Director, and is now a Thales Fellow. He started engineering in the 1970s, working on the design of novel optical systems for a wide range of system applications. He then worked successfully on increasingly diverse and complex systems, mostly for defence and aerospace applications, holding increasingly responsible positions in industry and government, running the UK MOD Integration Authority as an industrial secondee from 2005-8. He has been awarded 8 patents, and has published several papers at INCOSE international symposia (three of them winning Best Paper awards). He is a Visiting Professor at the University of Bristol, and a member of the BKCASE SEBOK author team. He was INCOSE UK Chapter president in 2004-6, was elected an INCOSE Fellow in 2009 and certified as an ESEP (Expert Systems Engineering Professional) in February 2010. He is a Chartered Engineer and a Fellow of the Institute of Physics.

REFERENCES

[1] Smith, General Sir Rupert, "The Utility of Force – the art of war in the modern world", Allen Lane, London, 2005

[2] 2006 Australian Defence Capability Development Manual.

[3] Stratton, James – valuable comments received on a previous draft of this paper, 2011.

[4] International Council On Systems Engineering UK Chapter "Z-3 Guide", 2009

[5] Hitchins, D K, "Systems Engineering – a 21st Century Systems Methodology", Wiley, 2007, p125,p129

[6] Sillitto, 2013: "Composable Capability" – Principles, strategies and methods for capability systems engineering – International Council On Systems Engineering, International Symposium, Philadelphia June 2013