


Composable capability –

principles, strategies and methods for capability systems engineering

Hillary Sillitto, CEng, ESEP, INCOSE Fellow

Thales Fellow

Acknowledgements also to James Stratton, SEIG, for insightful comments

THALES

Current systems engineering approaches for capability are difficult to understand, and not deployed consistently.

Why?

- Language is a barrier to understanding.
  - "there are more ideas in the human brain than there are words in the English Language"
- Top-down approach seems intractable.
- Mission thread approaches not widely espoused
  - infinite number of potential mission threads
  - risk of heavy investment in analysing situations that will never be encountered

Is there an alternative approach or way of looking at the problem that will give more benefit with less difficulty?

THALES

Purpose of systems architecture

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

If we do not "design" our operational systems architecture:

- *the capability as delivered by a set of interacting FEs will have behaviour that can be measured at that level but which may not be easy to predict.*
- *especially when the requirements for systems with the deployed force were specified in the absence of any understanding of what the composed system will eventually comprise.*

THALES

"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,
"The Utility of Force - the art of war in the modern world" - Allen Lane, 2005

Open systems benefits

There is convincing evidence that the open systems approach will reduce the cost and time of adapting to emerging threats and mission needs.

Illustrative RoI for Generic Vehicle Architecture (GVA)

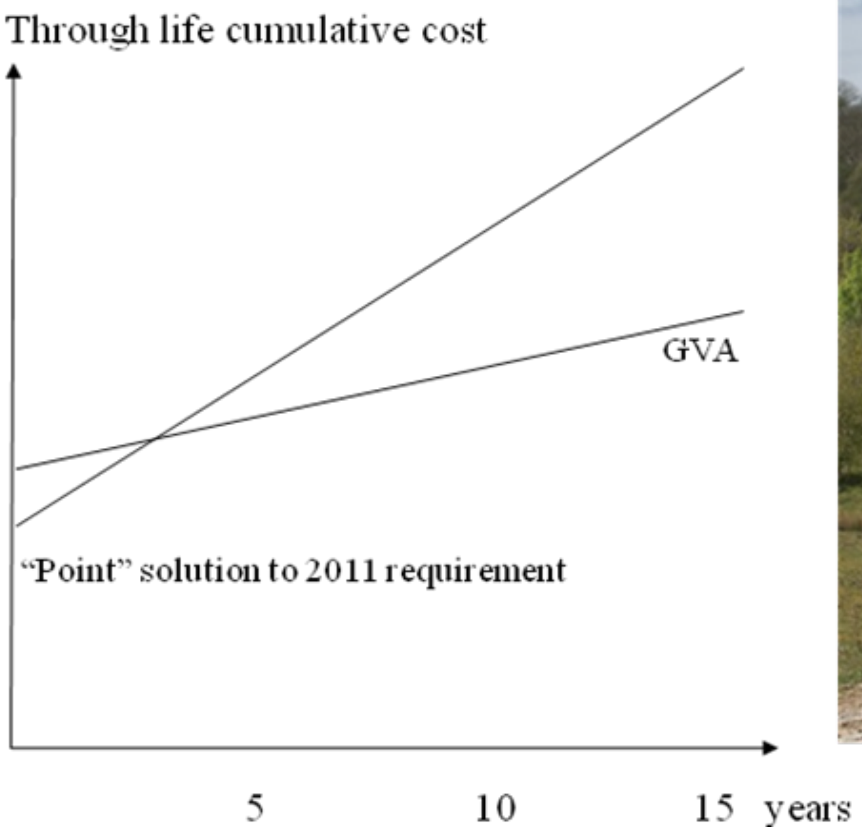


Photo by Force Protection Europe

Open systems are not a panacea!

The GVA approach does not give complete flexibility to do absolutely anything

- but it does allow us to do an awful lot of things very quickly and at low risk.

The concept of "plug and fight" is an oversimplification

- we can expect to "plug – configure – check – fight"
- and at a much higher tempo than is possible for closed systems, designed as "point solutions" to fixed requirements.

We will still need a design authority regime, to:

- maintain configuration rules
  - (for technical, procedural and operational aspects);
- maintain verification and validation capabilities - to check for
  - correct function, behaviour and performance,
  - unintended emergent properties;
- provide formal release to service
- manage configuration of the deployed fleet.

THALES

"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*".

[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 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.

It seems difficult!

THALES

Interoperability isn't new!

It is more about procedure than about computing:

- Guderian accurately described NEC before WW2 – "Achtung Panzer", 1937

In 1944-45, tank columns advancing through NW Europe got accurate and effective air support within a few minutes using

- "cab rank" air component organisation
- VHF voice radios,
- smoke flares
- well-defined, straightforward procedures

In the 1970s, the technology, doctrine and procedures were established for using laser targeting and laser guided ordnance.

- Ever since the 1980s it has been claimed that GPS based guidance systems made the demise of laser guided ordnance inevitable and imminent;
- But the technology has unique advantages and is still going strong.

THALES

Why did interoperability get so difficult?

Advent of digital communications and information systems on the battlefield:

Focus moved

- from procedural approaches for inter-unit interoperability
- to technology approaches for inter-equipment interoperability.

At the same time, MOD's acquisition organisation introduced "IPTs"

- sometimes referred to as "independent project teams",
- independence led to a plethora of "systems" (mainly equipment) being procured with minimal co-ordination.

The Integration Authority was set up to get some level of control over this:

An extreme challenge to get to grips with

- interactions between 200+ concurrent, asynchronous equipment procurements,
- without full access to information on in-service legacy equipments.

THALES

With benefit of hindsight:

Technical interoperability not needed between every possible combination of equipments.

But -

- with almost complete decoupling between the procurement architecture and the operational force structure,

it was quite difficult to establish:

- which systems did need to interoperate with each other,
- and under what operational conditions.
- and for what purpose,

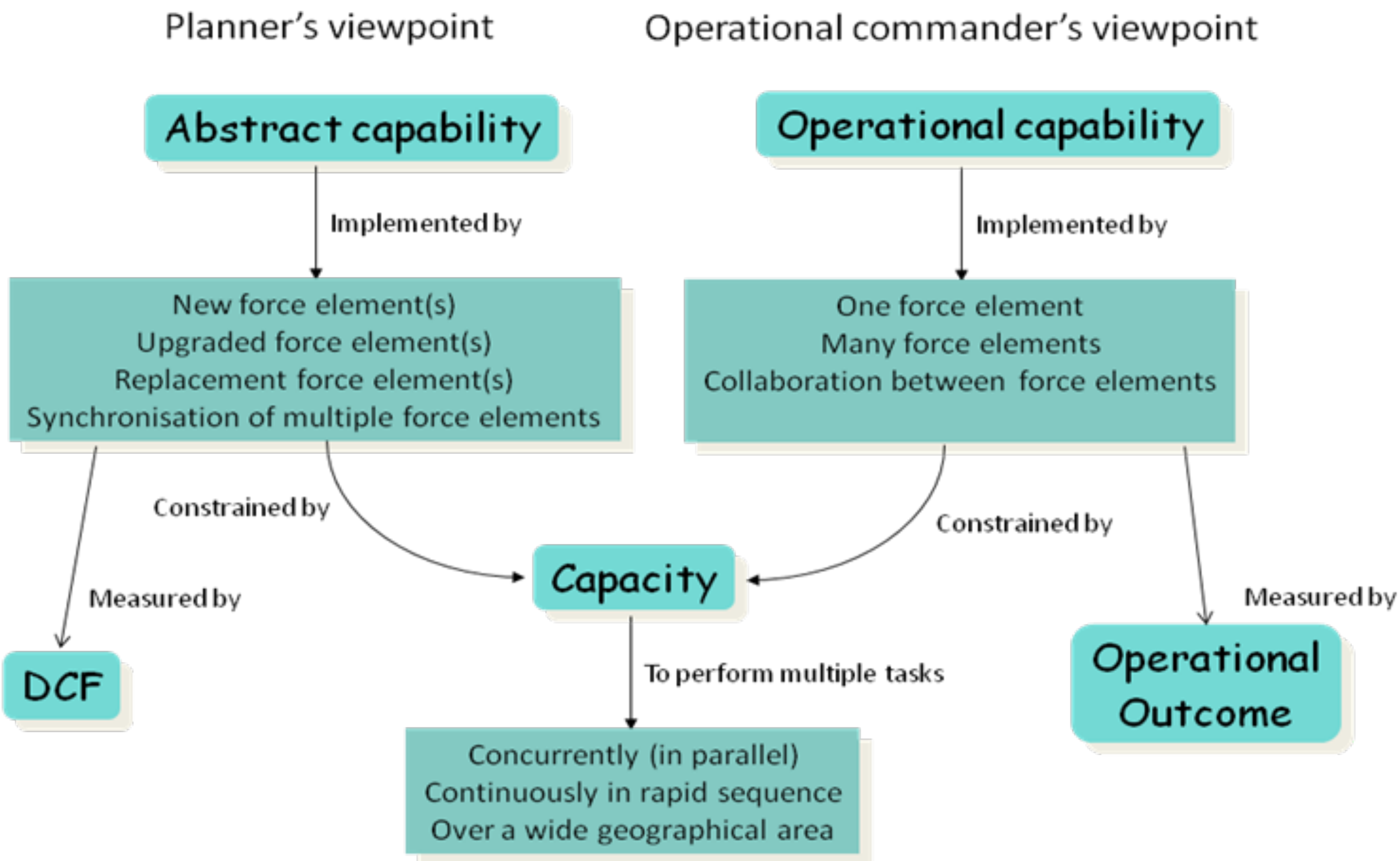
So:

Is there a different way of thinking about specifying defence systems?

- Focus on “purpose”, i.e. what constitutes mission success,
- “Force Element” not “Equipment”, is “system of interest.”

THALES

2 kinds of capability??



Systems engineering works for any “system” at any level

Principle for the engineering of complex systems, particularly those involving software (e.g. GVA):

- loosely coupled **objects**
- exchanging well-defined **services**
- for various **purposes**.

Compare this with previous definition of interoperability:

- 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.

Basically the same!

- We have well proven patterns and methods for doing this
- in both systems engineering and military domains

Key message - abstract system concepts, tools & methods are:

- scaleable and re-usable at different levels
- applicable to any type of system - technical, process, organisational, societal - -.
  - This gives systems engineering techniques huge power,
  - and huge potential for ambiguity and confusion!

THALES

Cost of change is lower earlier in lifecycle

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

Good architecture reduces cost of "likely change"

Critical success factors:

- Good modular design
- Good choice of interface points
- Good choice of interface standards
- Good choice of "chunk size" or system granularity

These choices are often set in or constrained by CONOPS.

- Cost of change in Concept phase even less – 0.1??
- Involve industry early better to understand cost/risk/capability tradespace

THALES

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

- basic behaviours,
- key parameters,
- timing and accuracy,

and also in terms of concurrency,

- so that we can know whether a force element will have to abandon other tasks to participate in a system of systems mission thread.

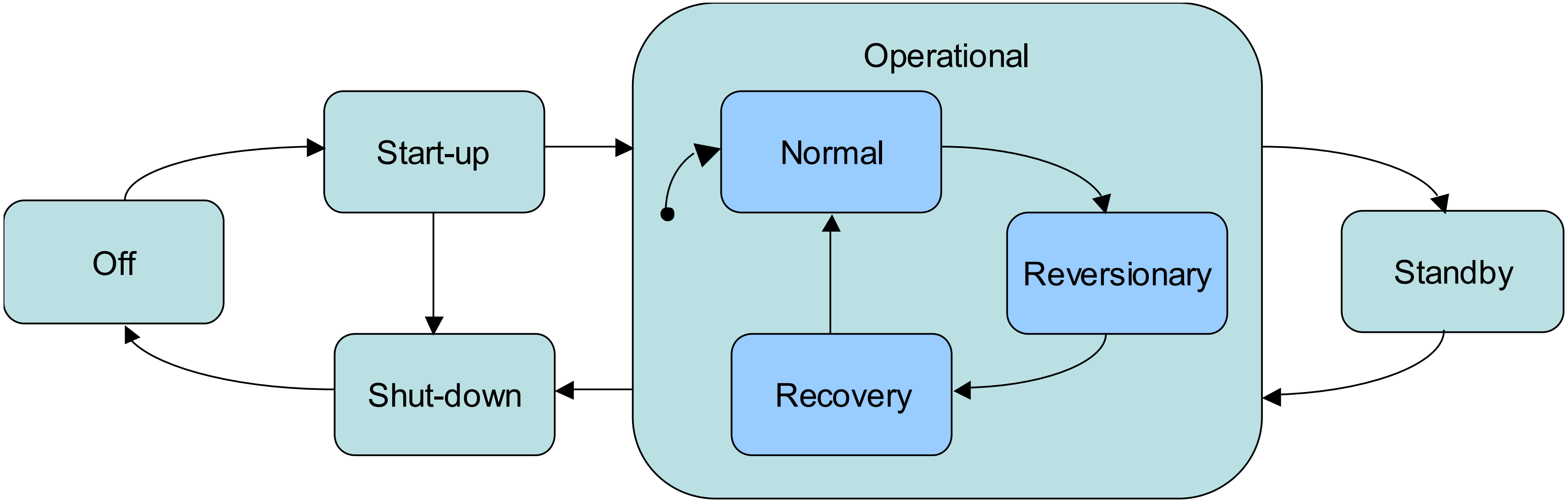
The way to invoke SoS functions is defined by the Sequence Diagram.

- The State model defines available concurrency.
- The operational rules model defines the behaviour under overload conditions.

THALES

Describing behaviour - System state (SV-10b)

The state behaviour of a system defines available concurrency between system functions, services and capabilities.



An "operational state model" (OV-6b) would look just the same. (Is that good or bad?)

THALES

Making it work in practice

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 now

- built from available units,
- flexible “task organisation” to deal with the unexpected and unforeseen.

What are the MOEs for “composability?

- time to restructure the force for a new mission?
- how well the capabilities of the restructured force match the new need?
- Recovery to (improved?) base state after mission?

What level of granularity should we use to define force elements?

- too big, the “chunks” don’t reflect operational reality;
- too granular, the whole thing becomes unmanageable.
- Best compromise: define force elements at the lowest level at which they are likely to deploy and operate independently for any appreciable period.
  - Air and sea – platforms?
  - Land - typically company/squadron level
  - Some specialised functions deployed at troop/platoon level?

THALES

Key premise:

We can, if we are careful, apply systems engineering methods to "force elements as systems"

Generic System model for Force, Force Element and DLODs:

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 & communication protocols; Operational State; Operational rules inc RoE	Capability; Command orchestrates operational services	Measures of Effectiveness
Force Element	Scope DLOD elements Command relationships Systems interfaces Programme dependencies	Command & communication protocols; Operational State; Operational rules inc RoE	Operational services (offered/ required/ exploited); Control synchronises technical functions	Measures of Operational Performance
Component of Capability (system or subsystem)	Scope People, process and product elements; Interfaces and Interactions	Command & communication protocols; System State; System rules	Technical functions Manage resources	Technical Performance Measures

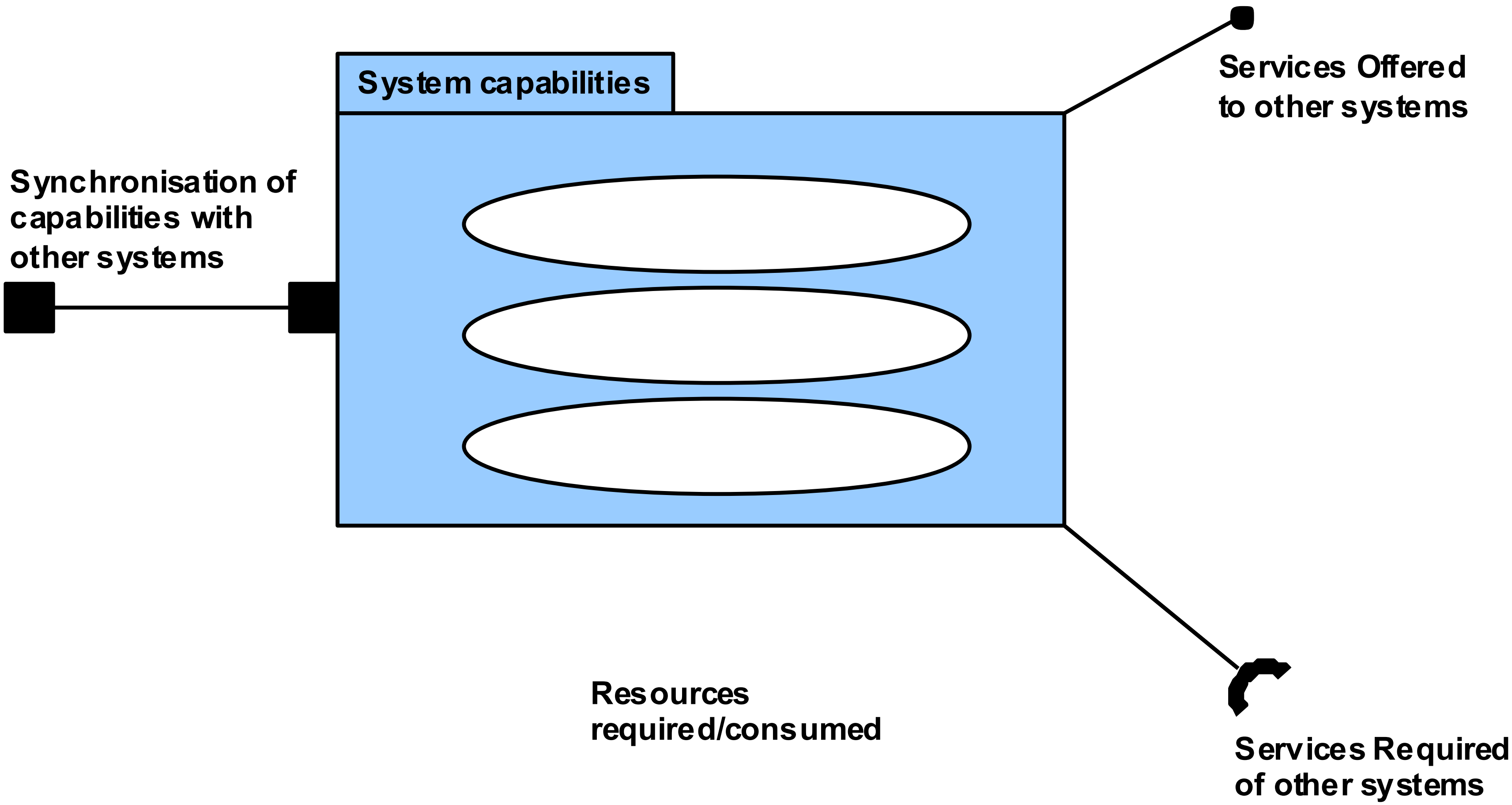
THALES

Generic system functional model looks very like the DCF!

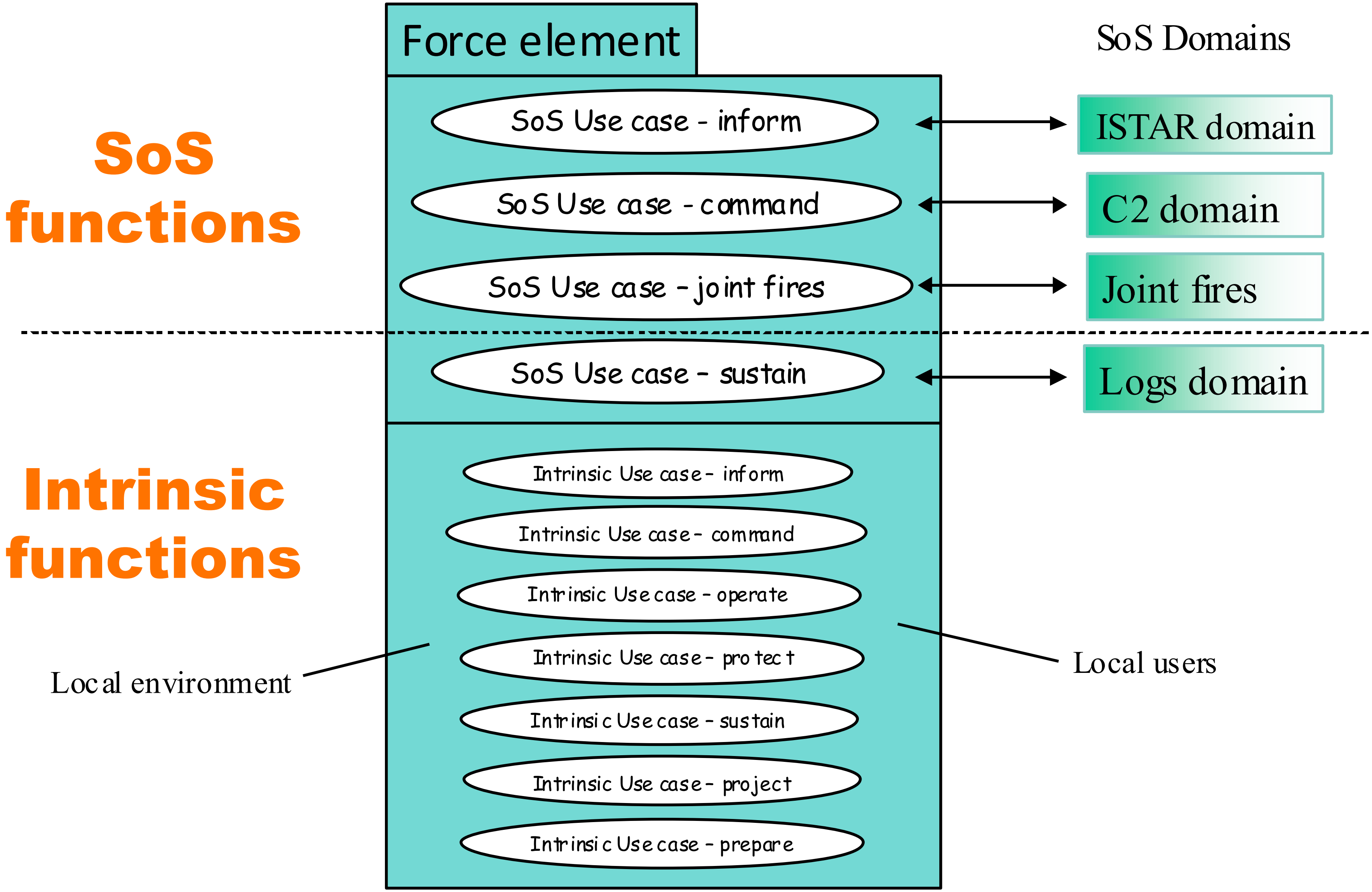
Generic	MOD specific	State
	Prepare	Non-operational
	Project	Transition
Sense	Inform	Operational
Manage	Command	
Operate	Operate	
Survive	Protect	
Sustain	Sustain	

THALES

"Service oriented pattern" works equally for a system element or force element



System of systems use cases must be designed, won't just happen!



In summary: layers of systems thinking in Defence

- Command orchestrates services across a force to create military effect
  - measured by MoEs;
- Force Elements synchronise technical functions to create operational services
  - measured by MoPs;
- Components of force elements (DLODs) provide technical functions/services
  - measured by TPMs
- Behaviour invokes state and functions
- States define allowable concurrency;

So we can define a standard template for a force element

This allows any force element to be fully specified and measured,

- Allows any mission thread requiring interaction between force elements to be analysed, validated and optimised.
- Allows doctrine for new capability configurations to be quickly established;
- 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.

THALES

Being more modular will allow MOD to be faster and more responsive in its acquisition.

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 make the individual force elements, the more freedom operational commanders will have 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, specify, measure and adjust the effects achieved by the interactions between force elements

THALES

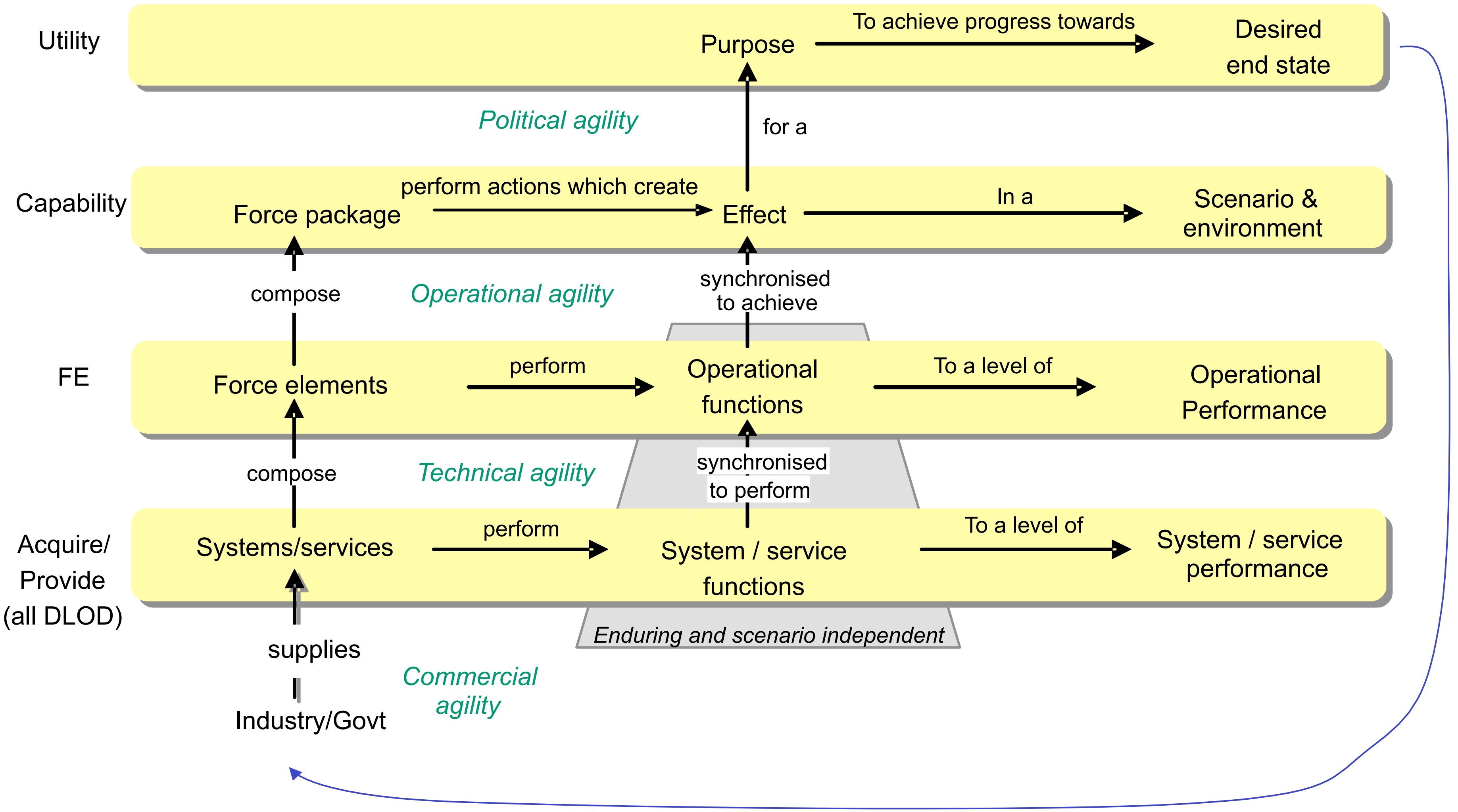
So: it's all about synthesising emergent properties!!!

Purpose – Context - Conops

THALES

Synthesising capability from force elements

Sillitto, INCOSE IS09, Singapore, July 2009






Any Questions? Arguments? Brickbats? Better ideas?

THALES