

Systems Methodology

Derek K Hitchins

Retired

Salisbury, UK

prof@hitchins.net

Abstract

The paper presents a “Systems Methodology” for creating optimum system solutions to complex problems and issues. The Holy Grail of systems engineering, a generic systems methodology has been the subject of the author’s ongoing research for over 20 years. The Systems Methodology, its precepts, concepts, ideas, methods and tools are presented in the paper, alongside a worked example showing the creative design of a sophisticated land-based defense capability.

N.B. An expanded description of the full systems methodology can be found at <http://www.hitchins.net/SysMethodology.html> and of the worked example at <http://www.hitchins.net/LandForce2010.html>

Background

The need for a systems methodology was perceived in the second half of the 20th Century.

Systems engineering was seen as a powerful method for solving complex problems, particularly in respect of major projects in the space program and the defense program. These early successes were based on systems science and system methods that were themselves relatively new: they had emerged in response to perceived limitations in the hard sciences, notably their inability to explain life, teleology (goal-seeking behavior), and the counter-intuitive behavior of “wholes,” or *gestalt*.

Systems methods concerned themselves with the synthesis of “whole open systems”

and with emergence, the latter caused by interactions between the parts within a system. Such methods, although effective, seemed alien and arcane to engineers concerned with more conventional methods for creating tangible and material end products, based on Cartesian reductionism.

The need was envisaged for a systems methodology that was accessible to engineers along with other disciplines from the applied and life sciences, so that the whole process, from addressing the problem to creating the optimum solution, could be understood and pursued with both rigor and expediency.

That need is even greater today, as our world continues to become more complex - as predicted by the Second Law of Thermodynamics.

The Systems Methodology (SM)

What is a systems methodology? Arthur D. Hall III (Hall, 1989), a founding father of modern systems engineering, put it like this. “Has mankind evolved to a point that there exists, or that with creative additions and re-combinations of modest proportions, there can be shown to be available, a common *systems methodology*, in terms of which we can conceive of, plan, design, construct, and use systems (procedures, machines, teams of people) of any arbitrary type in the service of mankind, and with low rates of failure?”

Hall was convinced that such a generic, problem-solving systems methodology was, indeed, within our grasp.

A methodology, or praxeology, consists of a process executed by skilled people using

methods and supported by tools. A systems methodology, then, is fundamentally a process incorporating system-scientific methods, supported by system thinking and simulation tools, undertaken by people with suitable systems and applied science skills.

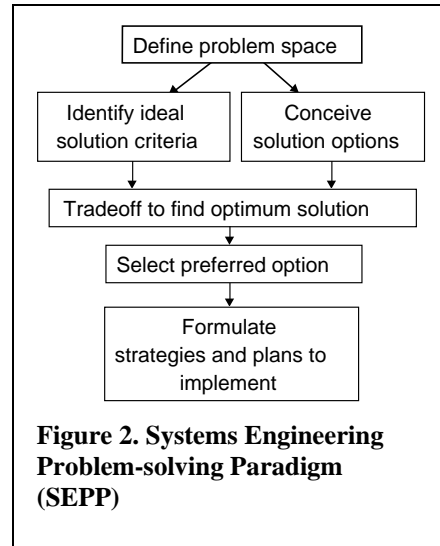
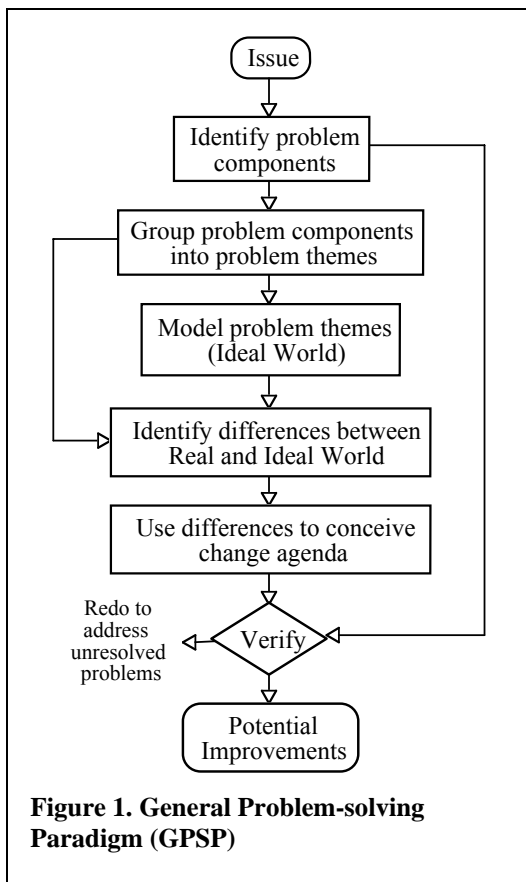
If we define systems engineering, not unreasonably, as follows, then *systems methodology* becomes the “*how*” of systems engineering:

“Systems engineering is the art and science of creating optimal system solutions to complex issues and problems.”

Problem-solving Paradigms

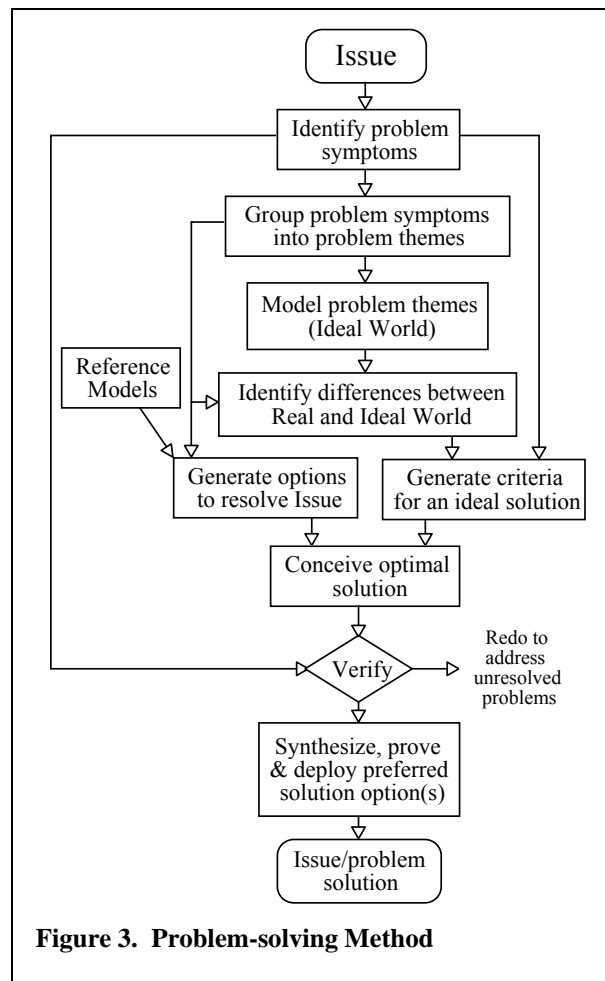
There are several well-known methods for solving problems. One is the so-called General Problem-solving Paradigm, (GPSP) – see Figure 1.

The figure, which is self explanatory, emphasizes the problem, or issue, domain. It



develops the idea of there being some Ideal World towards which we aspire, and of the real world; the difference between these two constitutes the driving force for change.

In contrast, the well-known systems



engineering problem-solving paradigm (SEPP), Figure 2, emphasizes the solution domain.

Joining these two paradigms offers the basis for providing system solutions to problems—see Figure 3. The process indicated in the figure has some key characteristics:

- It addresses all the problem symptoms together, and therefore the whole problem or issue. To address only part of the problem is to risk counter-intuitive behavior from the part solutions, which may make matters worse. (Forrester, 1975)
- The solution is synthesized without Cartesian reduction to avoid the risks

inherent in separate part solutions.

- There is a verification mechanism for ensuring completeness

Figure 4 shows how the top level Systems Methodology, based on the paradigm of Figure 3, presents as a behavior diagram. The centre column shows process or function. The left hand column shows what is needed by way of inputs, methods and resources to vitalize each function. The right hand column shows the output from each function or process.

The various methods are context, scale, type and solution independent (Hitchins, 1992, and Hitchins, 2003), so can be applied universally, as will be illustrated in following paragraphs.

paragraphs.

Figure 4 is still at high level, and so does not explain how these methods and processes are implemented; further elaboration is needed for that.

Following sections will expand on each step in the Systems Methodology. To illustrate in context, the SM will be used to find a creative, innovative solution to a particularly complex, but entirely fictitious, problem.

Issue and Problem Space

The Washington Business Herald Times, April 1st, 2004, carried an article indicating that US Defense has an apparent hole in its capability. Around the world exist vast areas of desert, open savannah and tundra where operations with current military vehicles were too difficult,

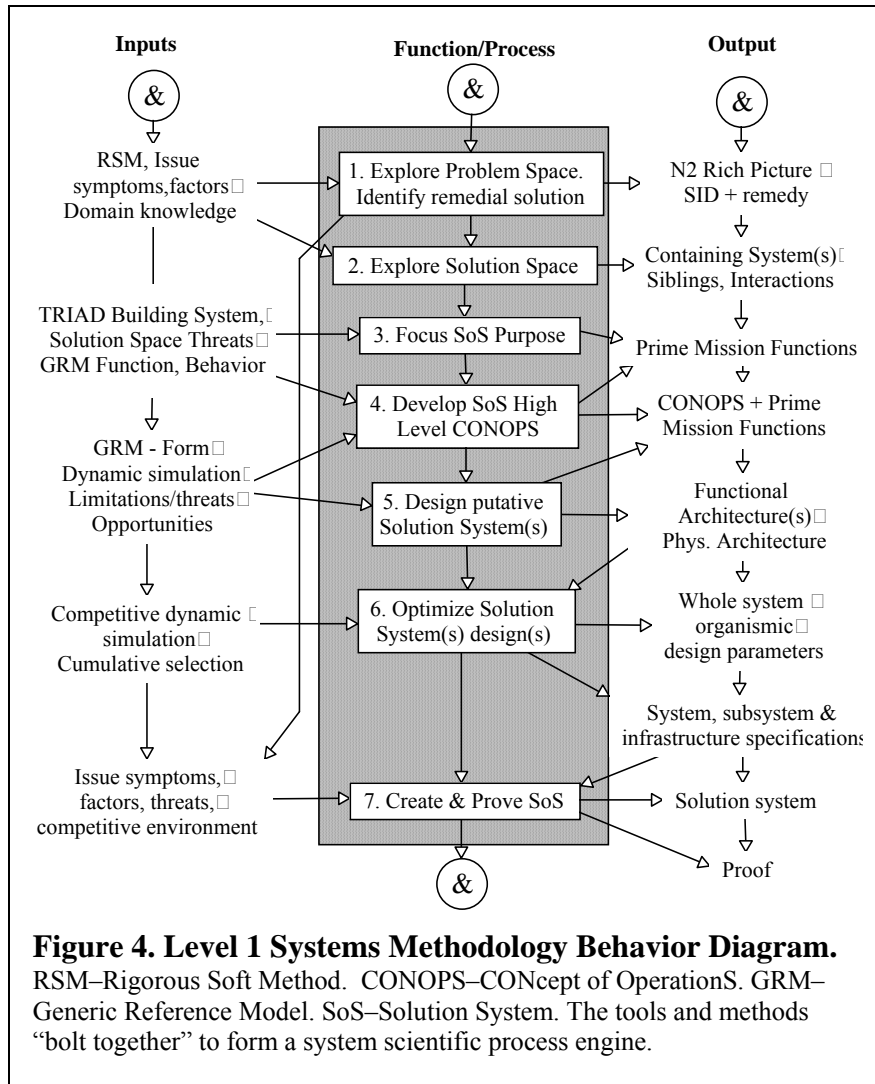


Figure 4. Level 1 Systems Methodology Behavior Diagram. RSM—Rigorous Soft Method. CONOPS—CONcept of OperationS. GRM—Generic Reference Model. SoS—Solution System. The tools and methods “bolt together” to form a system scientific process engine.

and/or where they would damage sensitive ecologies.

Communications were difficult in such areas, reliabilities were poor, and land forces were faced with situations more akin to naval operations at sea.

The article observed that DARPA, not noted for its altruism, was currently offering \$1M US for the first robotic crossing of 200 miles of the Mojave Desert in less than 10 hours.

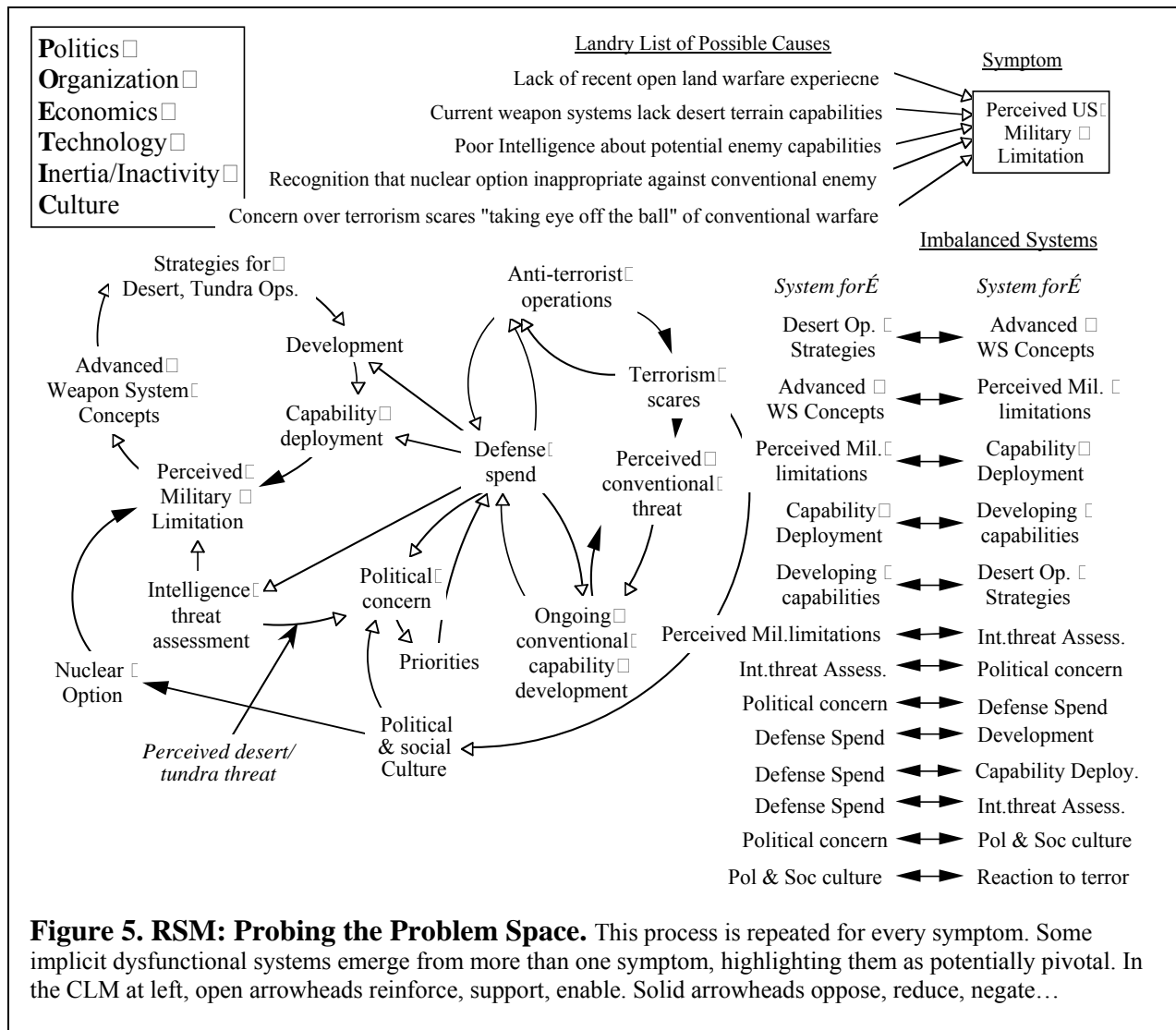
The (supposed) article observed that the US public also had a marked aversion to US military casualties.

Putting the various factors together suggested to the journalist that US Defense was researching into the use of robotic

vehicles to constitute some global defense capability.

Within the article there were a number of symptoms relating to the supposed issue:

- Perceived US military limitations in open land warfare
- Implied robot vehicle solution
- US political issue with casualties
- Uncertainty over desert operations strategies—positional Vs. maneuver warfare
- Perception that existing weapon systems may be unsuited to desert operations including communications, visual sights, radar, etc.
- Perception of military land situation



- being akin to naval operations at sea
- Perceived threat likely to overcome financial inhibitions
- Political urgency to attain new capability

Figure 5 shows the first stages in the RSM: the Rigorous Soft Methodology. Each symptom is taken in turn:

domain experts propose possible causes of that symptom, forming the so-called Laundry List (LL) (Richmond, 2001) at top centre. Pejorative terms are employed intentionally to draw out a richer spectrum of possible causes.

A range of typical causal agents is shown at top left: the acronym POETIC directs the experts' examination.

Elements from the LL, being presumed related, are formed into one or more causal loop models (CLMs – Roberts, 1983) as shown bottom left, adding any factors necessary to complete the loop logic. Dropping the pejorative terms results in an Ideal World representation—c.f. Figure 3.

Finally, a table is drawn up, bottom right, showing a locus of implicit dysfunctional systems within the problem space.

The process is repeated for each symptom, and the implicit dysfunctional systems, and their linkages, are aggregated in an N2 chart, where configuration entropy is minimized to reveal higher-level dysfunctional systems. See Figure 6.

The groupings reveal the Issue

US Culture	Ret terror	1	B	1						
	P&S Culture	2	1	D	1					
	Pol Concern	3		1	F	1	1			
US Politics	Int Thr't Ass	4		1	E	1				
	Perc Mil Lim	5			1	C	1			
	Def Spend	6		1		G		1		
Feasibility Constraints	Hum Perf	7		1	A		1			
	Des Cp Strat	8				O	1			
	Area cover	9				1	N	1		
Future Weapon Systems Ch'ics	Capab Deploy	10			1	1	M	1		
	Dev Capab	11		1			H	1	1	
	Adv WS Conc	12					1	J	1	
	WS perf	13				1	1	L	1	
	WS reliab	14					1	I	1	
	Log Tails	15						1	1	K

Figure 6. N2 Chart Showing Issue Themes. The N2 tool used to minimize configuration entropy uses abbreviated names for the dysfunctional systems. E.g., WS perf = (system for developing) weapon system performance. See Figure 7 for full titles.

Themes; while the detail from the CLMs, including linkages, is still preserved within the N2 chart, the themes present a higher-level view—c.f. Figure 3.

The N2 chart is not the clearest representation of the Problem Space. Instead, it is used to create a so-called SID, or Systems Interaction Diagram: see Figure 7.

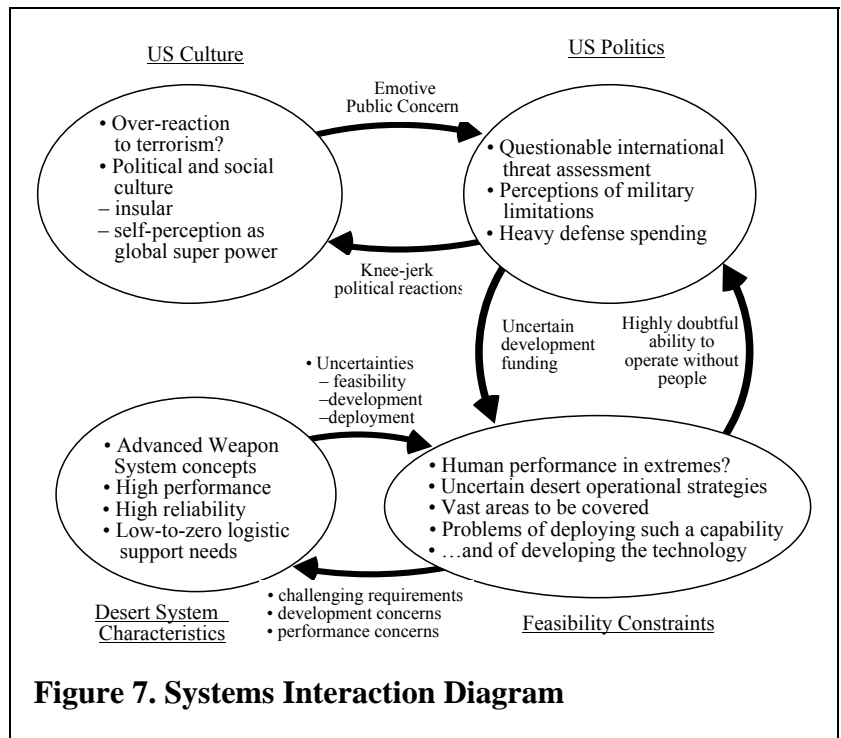
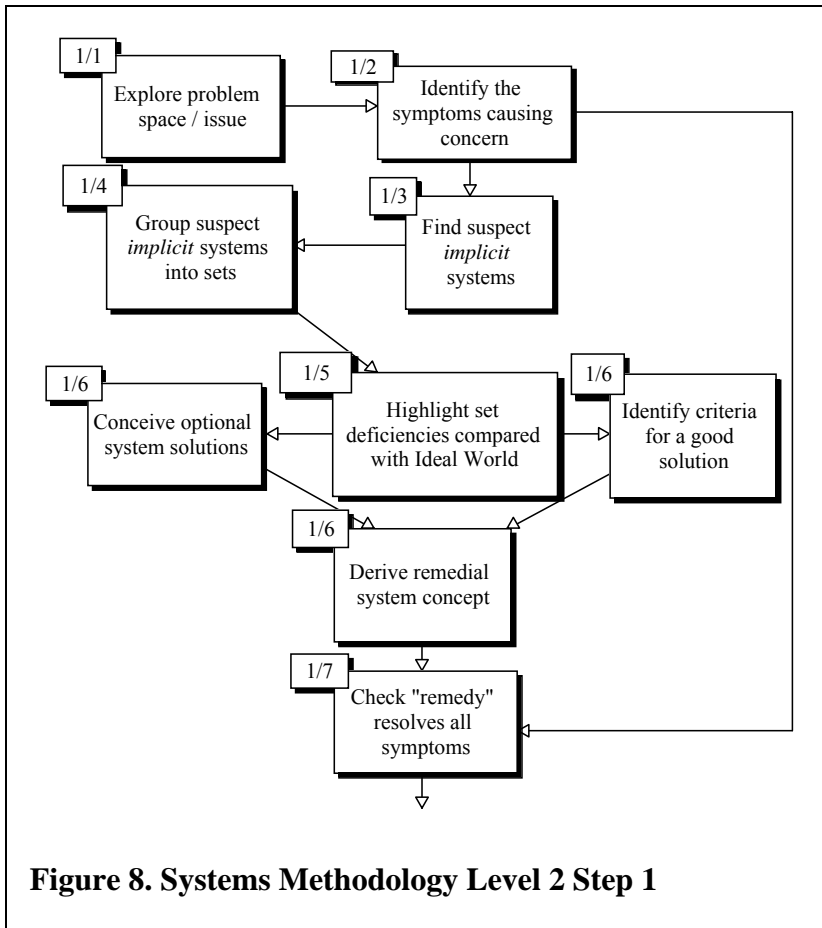
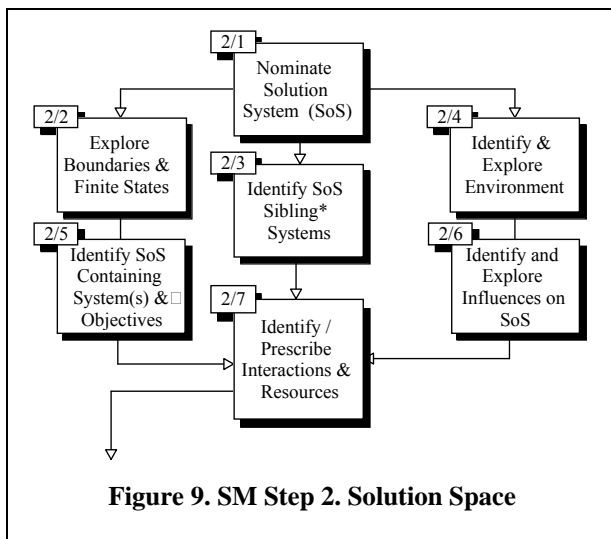


Figure 7. Systems Interaction Diagram



The SID has had pejorative terms re-inserted. In the process, it becomes almost like an intaglio, throwing into sharp relief the characteristics of some conceptual solution system, or systems.

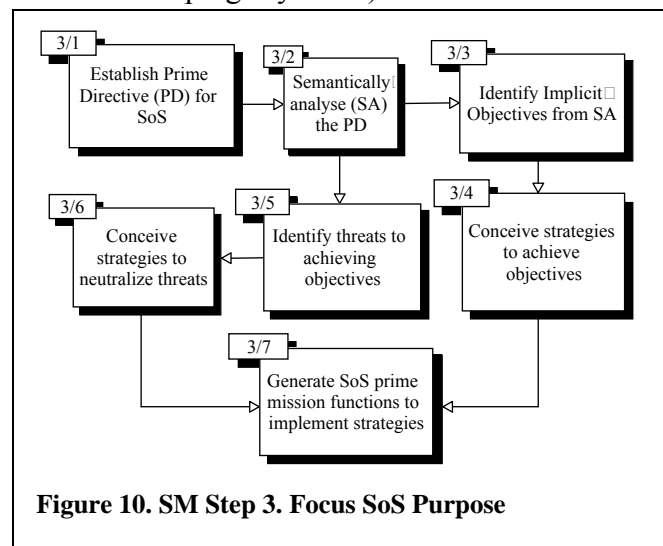
For instance, if a reliable robotic land



force could be conceived, designed and fielded quickly, it would resolve the issues displayed in the SID. The solution system is, as it were, an inverse projection from the SID. (There are other possible resolutions, including revised threat assessment, and changing political stances... we shall overlook these for the sake of example.)

Figure 8 shows a process model for the activities just presented. At this point, Step 1 of the Systems Methodology has investigated the Problem Space and shown the nature of a conceptual remedy. This is well short of a full systems solution, but it provides a start - often the most difficult step to take.

The second step looks forward towards the solution space: see Figure 9. Activity 2/1 names the solution system (SoS). Other activities look at the environment, within which the SoS will be operating, the interactions in which it will have a part, and the objectives of its Containing System (a UN Global Peacekeeping System?). The *value* of the



SoS will lie in the degree to which it contributes to its Containing Systems' objectives—along with its siblings.

Step 3, Figure 10, involves the use of the TRIAD Building System. In contrast to the RSM, which is a “soft” methodology for “messy” problems, the TRIAD Building System focuses on singular purpose.

Table 1 shows a typical Prime Directive in the first column and the word-by-word semantic analysis in the second column.

The simple ruse of identifying threats to achieving objectives, and then mounting strategies to neutralize those threats, turns out to be powerful. Solution system domain knowledge and expertise are needed to identify a full range of realistic threats.

The fourth step is the development of the SoS CONOPS, or concept of operations; this continues with the theme of establishing purpose, and along with it, functions within

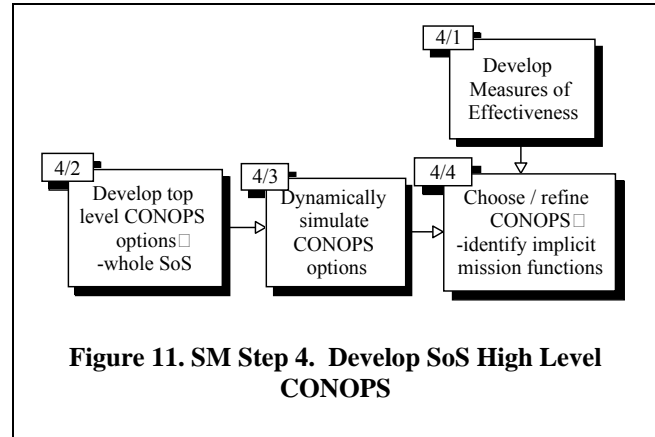


Figure 11. SM Step 4. Develop SoS High Level CONOPS

the SoS designed to achieve that purpose. See Figure 11.

There are many ways in which a CONOPS might be represented; in the example of Figure 12 below, a causal loop model (CLM) format has been chosen, although the subject matter is perhaps more procedural than causal.

Table 1. Operational Objectives, Strategies and Functions

Prime Directive 3/1	Semantic Analysis 3/2	Implicit Objectives 3/3	Strategies to achieve Objectives 3/4	Strategies (3/6) to overcome threats (3/5) to achieving objectives	Functions to support strategies 3/7
To neutralize enemies in... open desert and... tundra regions... ...around the world	To render ineffective... those identified by UN directive ABC as illegally entering, invading, existing and/or operating in... open, desolate, largely uninhabited tracts... and Arctic plains with permanently frozen subsoil, lichens, mosses, and dwarfed vegetation... wherever sanctioned by the UN	To deploy swiftly To move rapidly to scenes of incursion/activity To engage and deter, or overcome To identify legitimate enemies specifically To operate over wide areas, radically different environments, temperatures, going, etc. To operate within a UN mandate at all times	Air transportable Air deliverable High powered, high speed, all terrain vehicles. UMAs for remote identification and engagement where appropriate Vehicles to operate and fight on the move as an integrated unit, for speed, area coverage, avoidance of detection Fleet formation management to reduce enemy threat - open and tight, etc. Some vehicles to be self-steering, but under control of personnel in nearby vehicles /command posts	Pre-deployed cadre forces in area Some weapons/vehicles specialized for hot, wet, cold, ice, etc. conditions Use of non-lethal force to neutralize Use of UMAs to accelerate ahead of ground force Equipped: psy-ops, loudspeakers, leaflets, stun weapons, non-lethal anti-riot weapons Equipped: fuel-air and thermo-baric weapons (to warn as well as neutralize) + short-range electromagnetic pulse (SREMP) as non-lethal anti-technology weapon Equipped: canon, anti-tank missile, etc., anti-sniper lasers, enhanced remote ethnic/nationality laser identification	Cadre forces maintenance, communications and intelligence. Special vehicle support Lethal weapons training/practice Non-lethal weapons training and practice Fuel-air and thermo-baric weapons training/practice Human target identification Sniper location Real-time control of Rules of Engagement

This high-level, over-arching CONOPS identifies the main operational concepts; it will serve to correlate the separate and discrete CONOPS that will later be developed for each separate “unit” within the force, or defense capability. The sum of the discrete, unit CONOPS must, working together, synthesize the overarching CONOPS for the whole SoS.

The CONOPS brings together the strategies identified in the previous step. Note the UN roles of intelligence, alert and command; these presume that the US-led force will operate under the aegis of the UN. On alert, aircraft are loaded and launched, and the force is inserted into area; it joins up with a resident cadre force, and together they engage the incursors/insurgents. Reconnaissance employs satellite imagery and unmanned aircraft (UMAs) deployed by the force as it moves. The force continues to engage the incursors while being resourced and repaired as needed. Finally, the main force is extracted, leaving the cadre in area.

Step 5, Design, presents in two parts. The

first part concerns itself with the design of SoS “internal” function and behavior management. Use is made in Step 5 of the Generic Reference Model (GRM - Hitchins, 2003), a model representing the internal features of any system. It has three fundamental aspects: being, doing and thinking.

All systems have being, but need not have purpose, nor be sentient; the Solar System would be an immediate example. ‘Being’ is represented by the GRM (Form) Model, which is comprised of Structure, Influence and Potential.

Some systems also have, or appear to have, purpose and can do things; their “doing aspects” are represented by the GRM (Function) Model, which comprises Mission, Resource and Viability Management parts.

Yet other systems think, in addition to existing and doing... Thinking aspects are represented by the GRM (Behavior) Model, which is comprised of Cognition, Selection, Excitation and Belief System aspects.

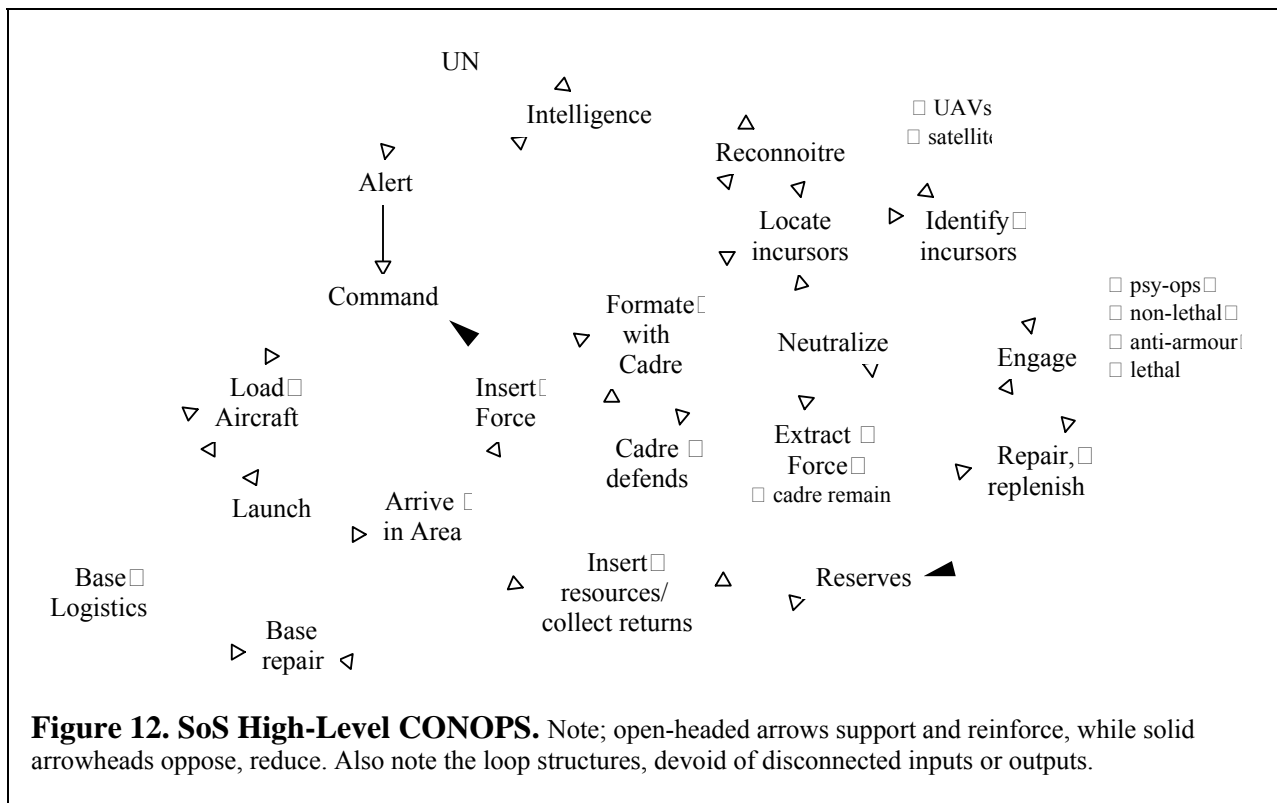


Figure 12. SoS High-Level CONOPS. Note; open-headed arrows support and reinforce, while solid arrowheads oppose, reduce. Also note the loop structures, devoid of disconnected inputs or outputs.

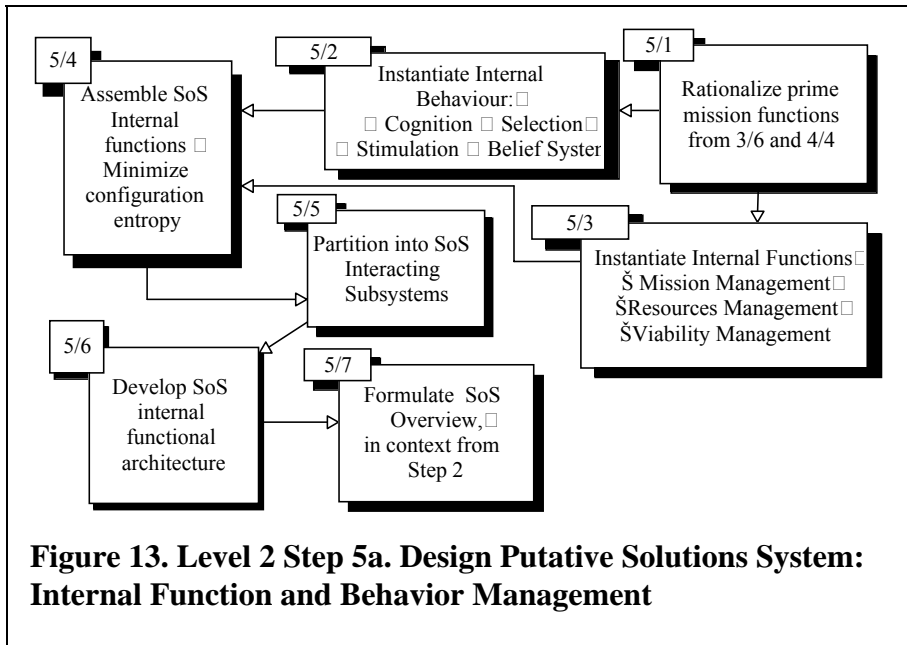


Figure 13. Level 2 Step 5a. Design Putative Solutions System: Internal Function and Behavior Management

The GRM as a whole can be used to represent any system: the manner of doing this will become apparent in the example. Figure 13 shows the process. Using the CONOPS and the prime mission functions identified in steps 3/6 and 4/4 as triggers, the internal mission management functions and behavioral

columns represents the instantiation of the generic function in the SoS. So the ‘Management of...information’ is achieved using a ‘communications center and an imaging center.’

Similarly, the ‘Management of... objectives’ is provided by ‘CPRM’ – Contin-

functions of the SoS are instantiated. This is most easily accomplished in tabular form, below.

Figure 14 shows the SoS instantiation of Function Management. It presents in three columns: one each for Mission Management, Viability Management and Resource Management.

Under Mission Management there are two further columns. The left-hand column identifies the GRM element, while the right-hand of the two

Internal Architecture Generation Table					
Mission Management <i>Management of...</i>		Viability Management <i>Management of...</i>		Resource Management <i>Management of...</i>	
GRM	SOI	GRM	SOI	GRM	SOI
-information	Com. centre Image Centre	-synergy	Formation management C2	...acquisition	CPRM Base Resupply Training
...objectives	CPRM	...survival	Formation management Self Defense System	...storage	Logistic support vehicles Ready use stores
...strategy & plans	Command & Control (C2)	...evolution	Performance Recording Systems	...distribution	Mobile Distribution Fleet
...execution	C2	...homeostasis	Climate control	...conversion	
...cooperation	C2	...maintenance	Mobile maintenance teams	...disposal	CPRM

Figure 14. GRM(Function) Instantiation

Internal Architecture Generation Table					
Cognition Management <i>Management of...</i>		Selection Management <i>Management of...</i>		Stimulation Management <i>Management of...</i>	
GRM	SOI	GRM	SOI	GRM	SOI
...tacit knowledge	Desert & tundra combat experts OJT	-nature	Psychological monitoring ... counselling	...motivation	Command and Control
...world models	Maps, satellite imagery, cultural perception	...experience	"Simulate before activate" practice	...activation	Command and Control
		...constraint	Rules of Engagement Discipline		

Figure 15. GRM (Behavior) Instantiation

agency Planning and Resource Management: (a typical function found in command and control centers.)

Figure 15 repeats the GRM instantiation process for SoS behavior, omitting Belief Systems.

All of the functions so far generated are assembled in an N2 chart that also represents

their mutual interactions, the configuration entropy of the set is minimized (by clustering) and the result drawn out as a functional architecture. See Figure 16. (Clustering, and hence minimizing configuration entropy, may be achieved either by eye or, as in this instance, using an automated N2 tool.)

Combat	Wpns Man	1	S	1							
	In t/ Recce	2	R	1							
	UMA Man	3	1	1	Q	1					
	Self Def	4			1	P	1				
	Form ate Man	5			1	O		1			
C3I	Clim Contr	6			J			1			
	Eng age Sim	7				M	1	1			
	Can d C	8			1	1	N	1	1		
	Image Centre	9				1	L		1		
	CPRM	10			1	1	I	1	1		
	ROE Man	11				1	A		1		
	Mobile Sup	12					1	H	1		
	Comm Centre	13				1	1	K	1	1	1
Logistics & Transport	Log Supp	14					1	1	G	1	
	Perf Rec	15					1	1	C	1	1
	Air Tran sp	16						1	F	1	
	Base Resup	17						1	1	E	1
Human Resources	Train ing	18							1	1	D
	Psych Mon	19							1		B

Figure 16. Functional Architecture N2 Chart. C3I is command, control, communications and intelligence –a military executive management and control socio-technical system. ROE is rules of Engagement. UMA is unmanned aircraft, equivalent to RPV, or remotely piloted vehicle, in this context.

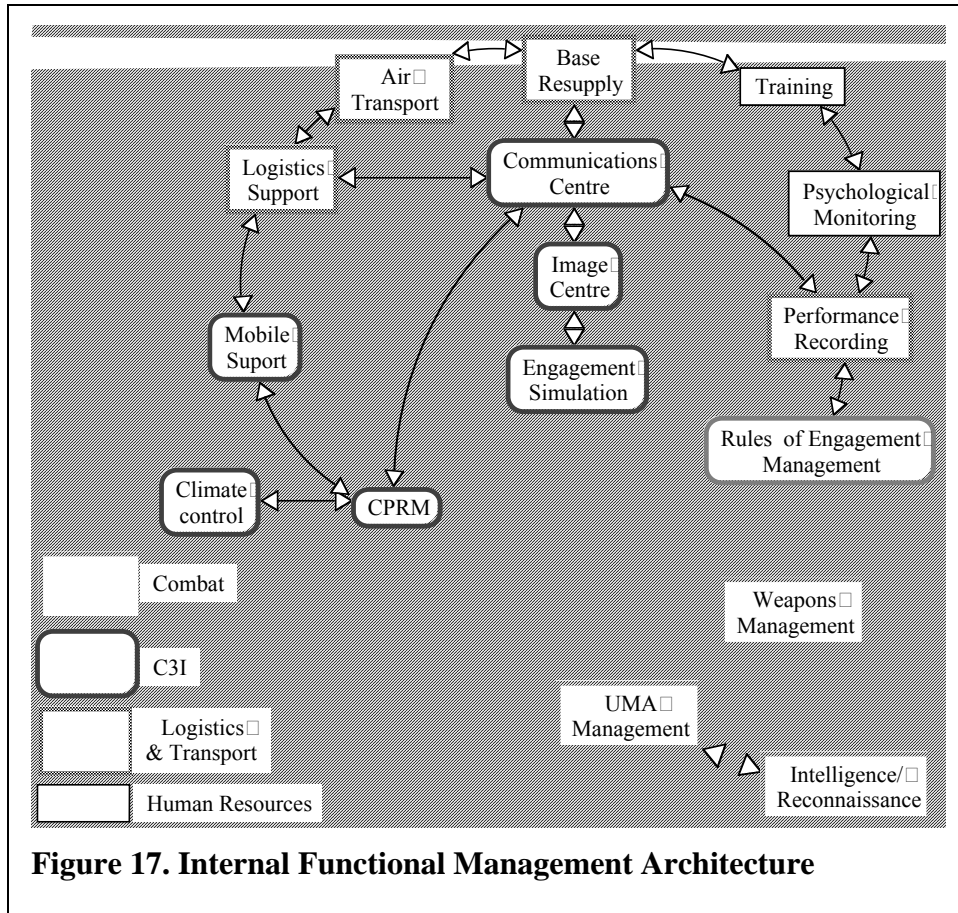


Figure 17. Internal Functional Management Architecture

Figure 17 (Step 5/6) shows the SoS functional architecture. Many of the strategies and prime mission functions are implicit in the figure.

For instance, Formation Management implies a set of vehicles on the move, changing formation to accommodate terrain or counteract threats.

Command & Control will concern itself with strategy and tactics. Weapons Management will be concerned with all kinds of lethal, area, non-lethal and soft-kill weapons: and so on. The internal architecture concerns itself with all the

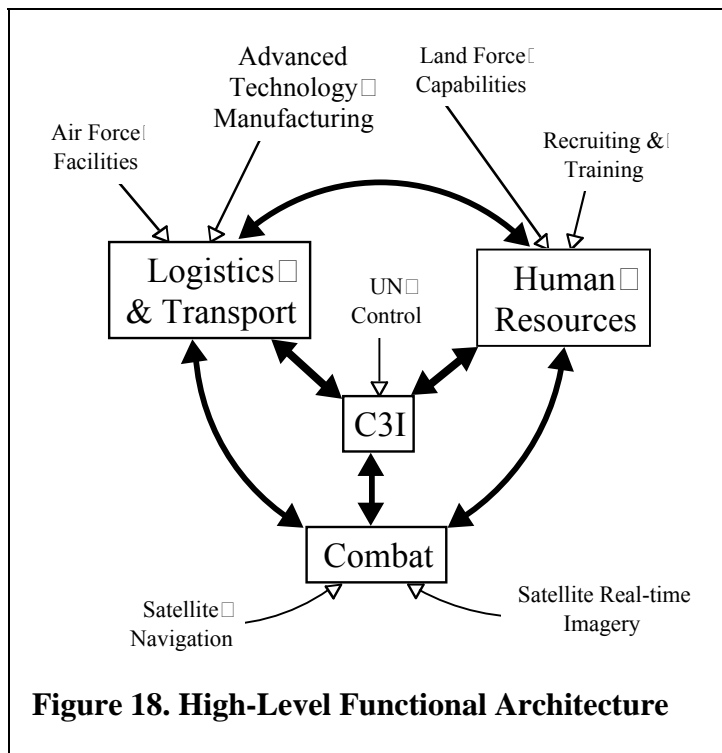


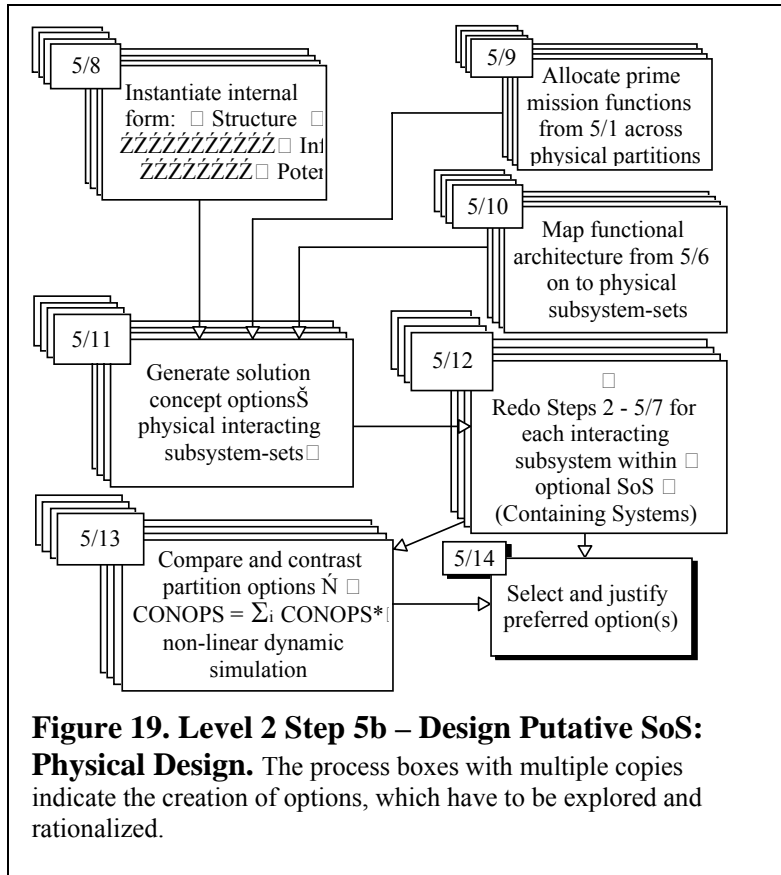
Figure 18. High-Level Functional Architecture

features needed within the SoS to pursue and execute the mission, while at the same time remaining viable and effective.

Figure 18 (Step 5/7) shows the various functions of Figure 17 synthesized into functional systems and interconnected with associated facilities, as identified in Step 2.

Every function, process and activity indicated and implied in these two figures is traceable back to the Prime Directive for the SoS, and thence to the original symptoms of the problem.

So, although the process is highly creative, that which has been potentially created is nonetheless logically traceable; there is nothing that cannot be justified. There are however, options that could have been considered. The use of



thermobaric and fuel-air weapons, for instance, might be thought of as somewhat unreasonable, even barbaric, particularly where there is concern for the preservation of fragile ecologies in desert and tundra. However, there is a wide range of weapons available and, as we shall see, there are steps yet to be taken to minimize environmental intrusion.

Figure 19 shows the Step 5b, the second part of SoS design. Only the GRM (Function) and the GRM (Behavior) have been used so far. The GRM (Form) identifies, *inter alia*, power and structure, and will accommodate all the physical subsystems: weapon systems, vehicle systems, UMAs, etc.

We are now able to posit optional (physical) solution concepts. Land Force 2010 (it has a name from Activity 2/1) could be formed around a land “carrier” able to launch and retrieve UMAs while on the

move. Other fighting vehicles/aircraft would be needed to defend this single carrier.

Or, it could comprise several vehicles, some able to launch, others able to retrieve, with yet others able to control, suggesting a functional split. Other fighting vehicles would defend the UMA core force.

Or, LF2010 could comprise a number of identical, semi-autonomous vehicles, each able to move, fight, launch, control and retrieve UMAs.

Once design moves from the functional to the physical, there are, potentially, so many options that there is a need for a way to generate the various optional arrangements, select the optimum, and demonstrate why it is the best solution.

One effective way to do this employs full simulation models, based on the GRM (see Figure 24 below), with genetic methods and cumulative selection to generate random configurations and test them under simulated operational conditions. In this way, a solution landscape containing hundreds of thousands of configurations can be explored, and the best solution system(s) evaluated and selected. The process generates the ideal functional/physical architecture and optimally maps prime mission functions on to physical partitions.

We do not have the luxury of space to describe these processes. To demonstrate the systems methodology at work, we will pursue only one of the many choices: a high technology, high maneuverability, bespoke option.

The limiting form factor in our chosen option is the capacity of the transport aircraft. We will assume a bespoke, V/STOL

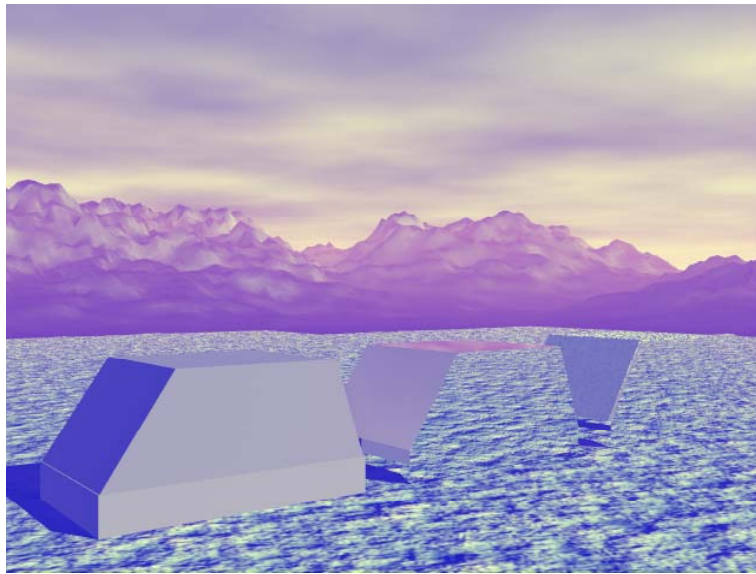


Figure 20. Chameleon Camouflage. At left, the simulated TLE is not camouflaged. The center, the TLE is mirrored, showing reflected blue sky on top. At right, the upper TLE surface shows a “photocopy” of the ground under the TLE, and so becomes invisible from above.

resulting force will be called a SWARM.

In this option, each of the TLEs is externally identical; each has a skirt which can be used to hover, to get out of bogs, ponds, quicksand, cross water, ice, etc; under the skirt are retractable drive wheels/half-tracks for normal road/off road use; there are no windows, doors, or visible apertures; the sides are covered with a material that can be induced to reflect like a mirror; the top displays a live “photocopy” of the road being passed over—see Figure 20

In this option, UMAs are designed to appear as indigenous birds and insects, both for camouflage and to minimize intrusion into sensitive ecologies.

The principal UMA is a Raptor. It is automatically launched from, and recovered to, a TLE. The Raptor also carries one or more semi-autonomous “Dragonflies,” which can get close to any action. Dragonflies are shorter range, can hover, transmit video and audio, deliver some weapons and can operate as Kamikazes to minimize collateral damage.

Without going into any more detail, we have effectively reached the first level of SoS Design, although it requires backing up with drawings, simulations and perhaps prototyping.

One major item remains: the network of communications necessary to enable the various elements to operate as one—so-called network-enable capability (NEC). This will be provided, in this option, by a DTDMA (distributed time division multiple access) system providing Communications, relative Navigation and Identification in the one CNI system. It will operate in an atmospheric absorption band to prevent detection,

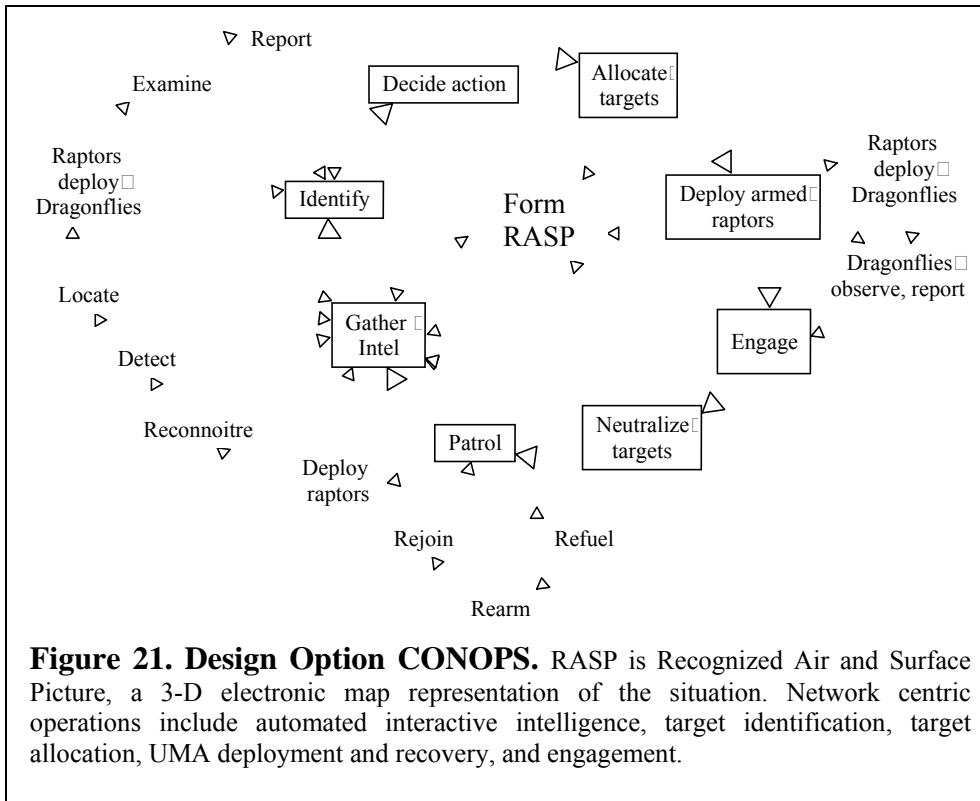
transport, 2,500nm hop range when fully loaded, carrying capacity at maximum range of 35 tons.

We will posit a 10-ton vehicle—much lighter than a tank—so that we may load three vehicles, or Transportable Land Elements (TLEs), in tandem per aircraft.

The remaining 5-tons in the V/STOL are command and control plus CPRM; remote vehicle control stations; intelligence suite; communications, including satellite communications; logistic supplies; and repair bays.

In this option, TLEs are not intended to fight. Instead, they carry a wide range of UMA/RPVs that can deliver weapons. So, operators are not intended to come into contact with incursors, to minimize casualties.

A full force might comprise 20+ such aircraft, with 60+ TLEs deployed at once, each with multiple UMA/RPVs active simultaneously, all on the go, in changing formation, adapting in real time. The



exchange ratio, casualty exchange ratio, return on capital employed (ROCE), or any combination according to system, situation and need.

Optimization of a complex SoS design requires that it be observed and measured in operation. Measures such as performance and effectiveness are emergent: they emerge from the interactions of all the many parts within the SoS. Altering any one

exploitation and compromise, and has sufficient capacity for video, audio, vehicle control, status, and all other network traffic.

The CONOPS for this particular option is shown in Figure 21. In any viable option, the CONOPS for the partitioned SoS would achieve the CONOPS for the whole SoS, Figure 11. In particular, the act of partitioning the SoS would not be allowed to degrade the prime mission functions and behaviors of the SoS.

Figure 22 shows the process for Step 6 of the SM – Optimizing the Design. Optimization is an essential step if the SoS is to be acceptable. A design may be optimized in respect of many different, or combined criteria. For instance, it may be optimized for: performance, efficiency, effectiveness, cost-effectiveness, cost-

part also alters its interactions with the others, so the outcome of even minor changes is not simple to predict. By successively altering various parts and

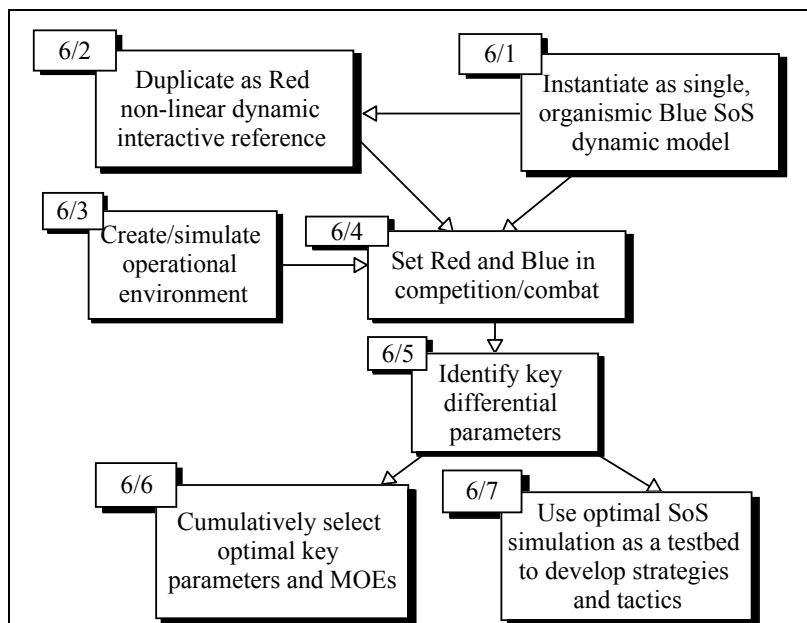
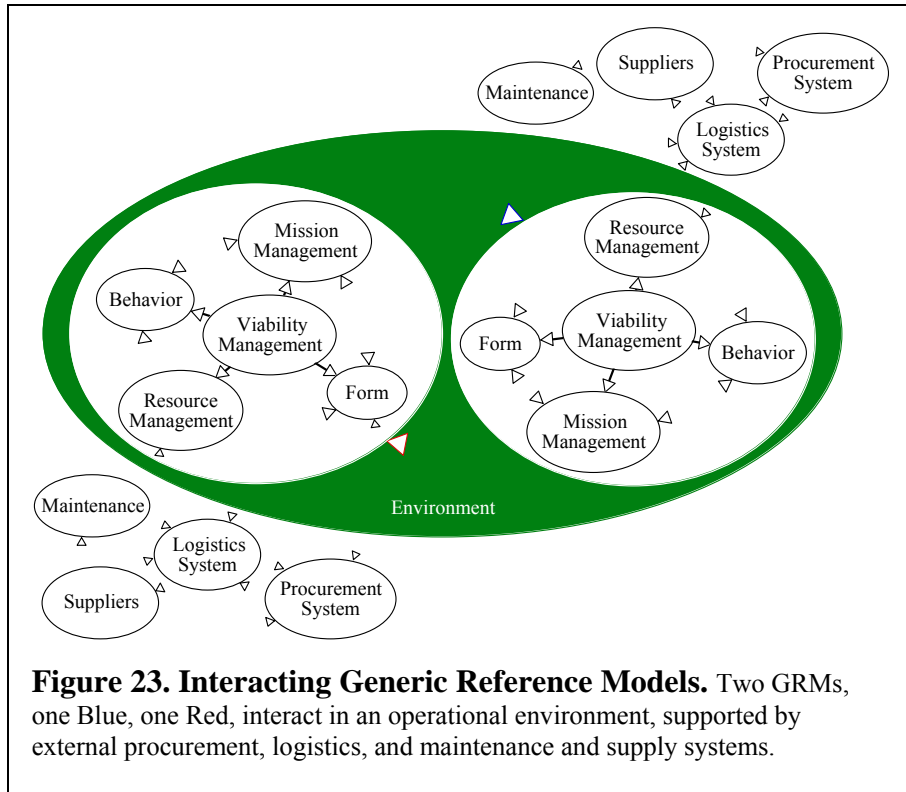


Figure 22. Systems Methodology Level 2 Step 6. Design Optimization



observing the emergent behavior of the whole, it is possible to progressively optimize measures.

However, The SoS as a whole also interacts with other systems and its environment. For instance, its outflows affect other systems, and other systems affect its inflows. So, optimum performance, for instance, can be identified only when the SoS is operating and interacting with other systems.

One way to get around this dilemma is to represent the SoS in dynamic simulation, interacting with another system. The second system could be in competition, or in combat. If that other system is a replica of the SoS, then the impacts of both inflow and outflows can be recognized and accommodated at the same time – see Figure 23 and Figure 24. (In our example, a Blue and a Red Land Force 2010 would engage in simulated combat on a variety of simulated tundra and desert terrains, 6/3).

Initially all features and parameters of Blue, 6/1 and Red, 6/2, are identical.

Combat results in a standoff, with both forces balanced. Each will inflict similar damage on the other, suffer similar losses, and so on.

Red is then held constant as a dynamic, interactive reference. Individual parameters in Blue are changed. Resulting changes in Blue *and* Red performance, effectiveness, etc., are observed, due to their mutual interaction. In this way, the often-complex behavior of the interacting systems can be observed, sensitive

parameters identified, and counter-intuitive behavior observed: Steps 6/4 and 6/5.

It is possible to build the simulations using “genes” to “code” for different features. So, in our example, there might be a gene coding for radar transmitter power, another for the number of Raptors carried on a TLE, another for the range of a Dragonfly, another for DTDMA network performance, and so on. In each case, it would be difficult, or impossible, to predict the effect of changing the parameter on conflict outcome.

Random configurations of Blue System, Step 6/1, can then be simulated and compared automatically, and cumulative selection methods can be employed to progressively enhance Blue’s capability in respect of Red; see Step 6/6.

This process optimizes the whole system while in operation, rather than trying to optimize the parts (V/STOL, TLE, Raptor, Dragonfly, etc.) individually. In consequence, it is significantly more powerful.

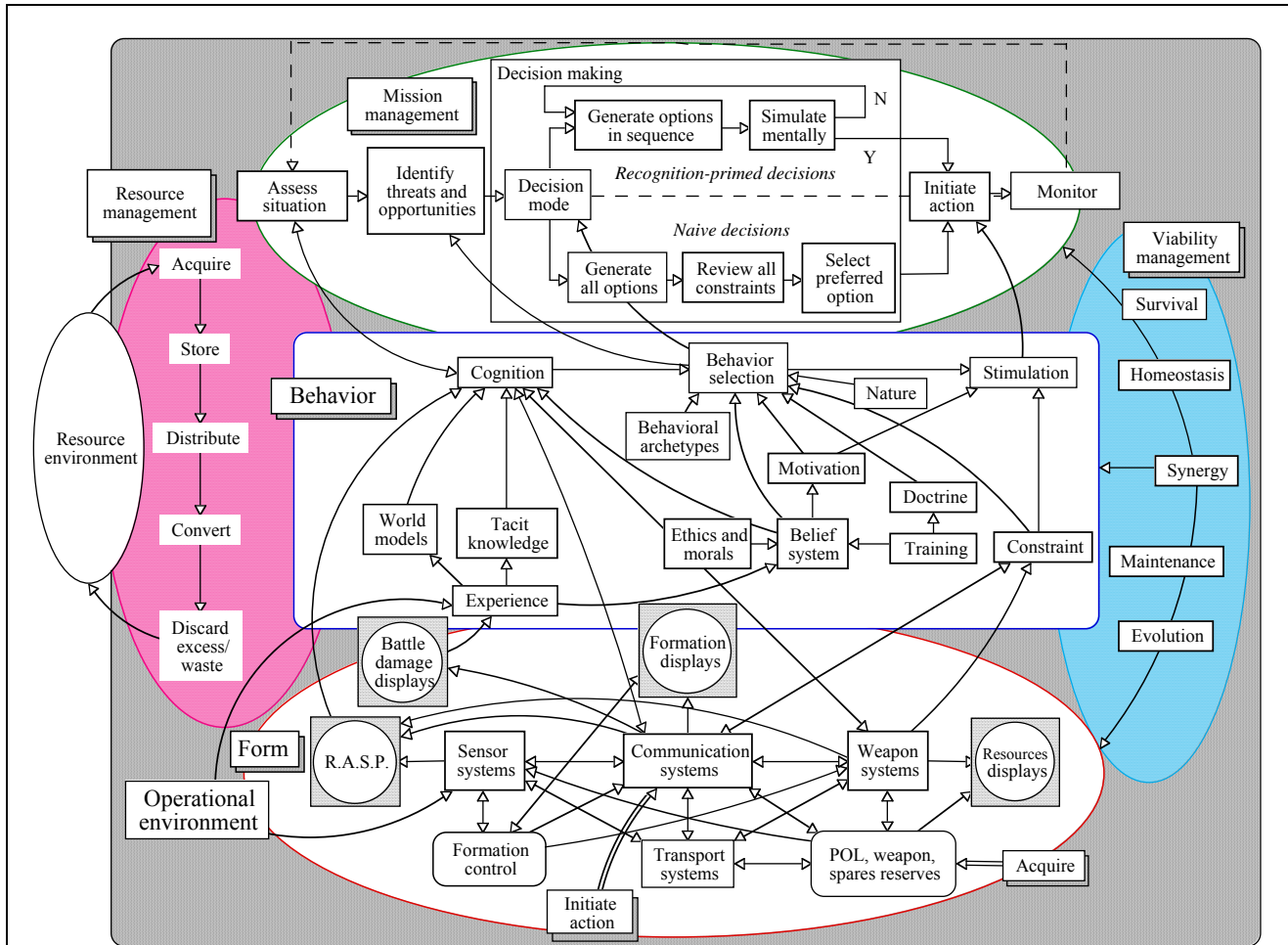


Figure 24. The GRM, in layered form, Instantiated as Land Force 2010. The diagram shows one half of pair of interacting Land Forces in combat. The bottom layer is the GRM (Form), which contains the technology, the physical subsystems, etc. The middle layer, GRM (Behavior), represents the people and their response to stimulus. The top layer and the left and right columns represent the GRM (Function): Mission, Resource, and Viability Management. Two such models, mutually interacting, offer basis for dynamic design optimization.

If, in additions to genes for functions and behavior, genes are included for partitioning, then this process will also subsume SM Step 5b, since configuration options will include partition/form options as well as function and behavior.

(SoS system design may be evolved further by testing it against a variety of (simulated) opposition in a variety of environments and situations.)

The genetic/cumulative selection processes are powerful: they can sift through an n-dimensional landscape containing thousands, if not millions, of optional

solutions in relatively short order. However, the process is conceptually similar to the SEPP of Figure 2, i.e., generating optional solutions and selecting that which best meets the criteria for a good solution.

The process model for the final SM Step 7 is shown in Figure 25 below. It is largely self-explanatory, especially when it is realized that some of the parts/subsystems of the SoS may be teams of people who require training, some parts may be available in the market, and some may need specific research, development and manufacture.

SM Step 7 is not unique: there are other

ways in which the SoS may be synthesized. However, treating each of the parts/ subsystems separately has the potential to create the solution system in the shortest sensible time. It also separates the risks, so that any one of the parallel projects that proves troublesome may be subject to additional support, without any other of the parallel projects being affected.

On the other hand, it would be essential to ensure that the developing part/subsystems did not “wander” from their specification during development such that their dynamic emergent properties, capabilities and behaviors (DEPCABs) and interactive characteristics change materially.

Each of the subsystems specified at 7/4 will itself be a system. The SM processes, Steps 2-7 may be repeated, therefore, for each subsystem. Step 2 will, of course, be relatively straightforward: the containing system will be the SoS, and the siblings will be the other subsystems forming the SoS.

This process of repeating Steps 2-7 shows that the SM may be used recursively, designing systems, subsystems, sub-subsystems, etc., until sufficient detail is generated to permit, e.g., specification, and the choice of appropriate technology.

There has been no mention of technology until this point. In the general case, the SoS need not employ any

technology; instead it might comprise teams of people, procedures, processes, organization, etc.

However, our example does present technological issues. SM practitioners would need to be technologically aware, so that they did not conceive and specify subsystems or parts that were infeasible.

The dynamic test environment, 7/6, which includes the original problem symptoms, is a means of ensuring that the created SoS, comprising the various subsystems brought together, tested and integrated, will resolve the original symptoms. The overall SM is therefore self-correcting: not only should it find a solution where one exists, but also it should not offer

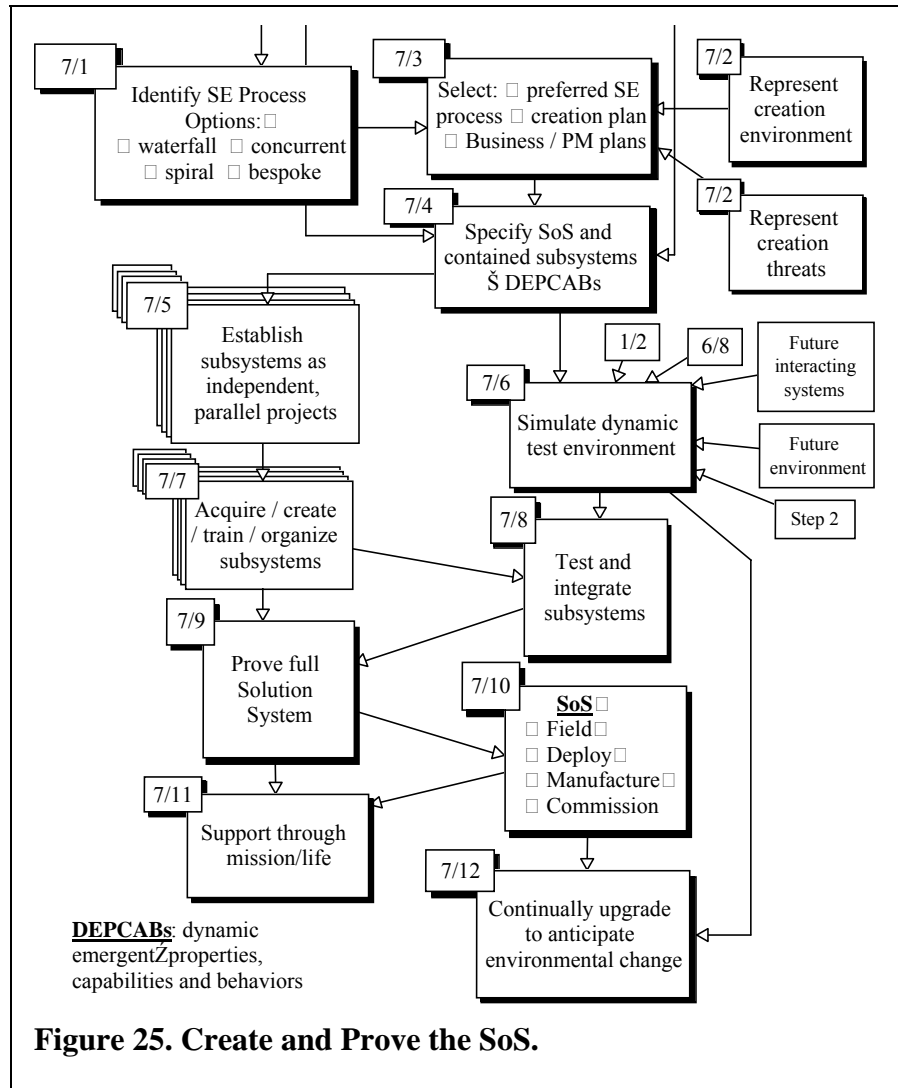
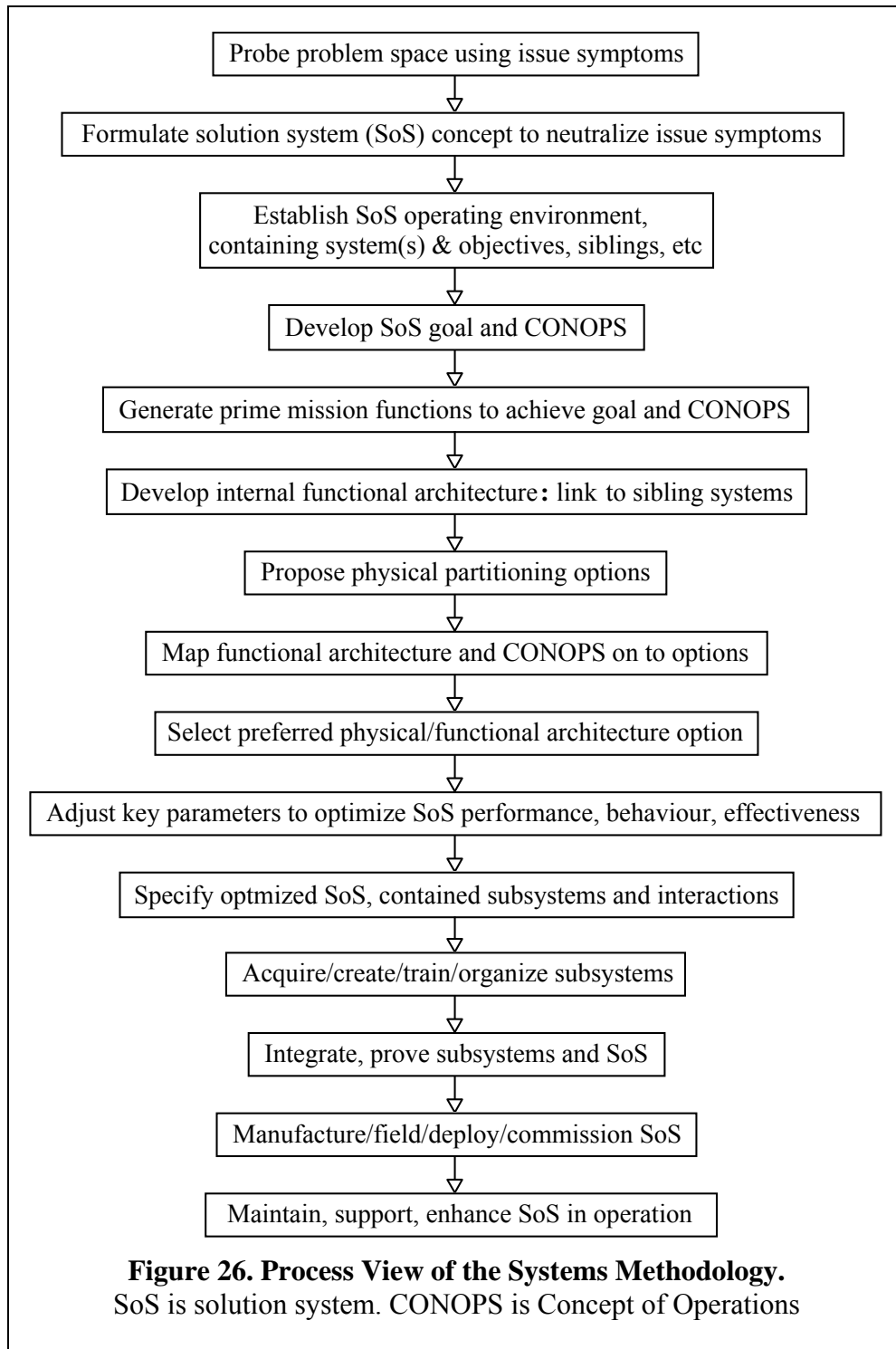


Figure 25. Create and Prove the SoS.



a solution that is invalid or incomplete.

Figure 26 shows a process view of the Systems Methodology, formed by logically interconnecting the various strategies and techniques that are embedded in the systems methodologies, methods and tools. The

process engenders synthesis, holism and the organismic analogy; the Systems Methodology is, itself, an open interactive system, and its processes can, and should, be subjected to test as both systems scientific and mathematically provable.

Conclusions

A systems methodology (SM) can be formulated that does, indeed, provide a route from complex problem to optimum system solution – as Arthur D. Hall and others believed it would in the 1980s. The SM:

- is context, system-type, system-scale and solution independent
- is science-based and mathematically provable
- may, therefore, be used to tackle a wide range of problems and produce an even wider range of solution systems
- addresses the whole problem, i.e., is holistic
- synthesizes whole solution systems without Cartesian reduction
- creates organismic solution systems
 - non-linear multipart systems that act as a unified whole
- optimizes solution system designs
- employs system methodologies
 - methods are both provable (Hitchins, 2003) and falsifiable (Popper, 1972).
 - could be improved upon for ease of use, tool support, etc.
- requires inputs from, and activities to be undertaken by, domain, operational and other experts.
- exploits simulation, and genetic methods,
 - trained and experienced practitioners should execute such activities with integrity
- is system-theoretic, traceable and logical, yet can be highly creative and innovative
- “conducts the creative process,”
 - that which flows from problem to solution is the organized product of rational human intellect.

- problem inspired
- optimal solution driven
- concept before detail
- purpose before function
- functional before physical

Reference

- Forrester, J. W., *World Dynamics*, Cambridge, MA: MIT Press, 1973
- Hall, A.D. III, *Metasystems Methodology*, Pergamon Press, Oxford, England, 1989
- Hitchins, D.K., *Putting Systems to Work*, Wiley, Chichester, England, 1992
- Hitchins, D. K., *Advanced Systems Thinking, Engineering and Management*, Artech House, Norwood MA, 2003
- Popper, K., *Conjectures and Refutations: the Growth of Scientific Knowledge*, London, England, Routledge and Kegan Paul, 1972
- Richmond, B., *An Introduction to Systems Thinking*, High Performance Systems Inc., Hanover NH, 2001
- Roberts, N., et al, *Computer Simulation*, Addison –Wesley, Reading MA, 1983

Biography



Derek Hitchins spent some 22 years in the Royal Air Force, 12 years in the defense industry, 10 years as an academic, and 10 years as a consultant—all in systems engineering of one kind or another.

He held chairs in engineering management, command & control and systems science. He was the inaugural president of the UK Chapter of INCOSE, and is an INCOSE Pioneer.

He is now semi-retired but continues with his research, writes books and lectures occasionally, particularly on his abiding passion for the original systems engineering in ancient Egypt.