Evolving Support Solutions

Evolution, not revolution: how the application of a simple systems approach can transform the effectiveness of an ILS programme.

A previous Aspire blog post, **"What is ILS"**, revisited the very basics of Integrated Logistic Support (ILS) to explore the questions:

- What is ILS?
- What is ILS for?
- What is it we're "Integrating"?

Too Soon… Too Late

The core problem addressed in this article is a perennial one for Support Engineers; that is that when they try to do the right thing, when they attempt to conduct support analyses, to carry out Through Life Cost predictions, or to optimise the design of the Mission System, for example, they will often be told that it is too soon for such activities. The argument being that the design is not mature and hence there simply is no appropriate data available.

The problem is that "**Too Soon**" can become "**Too Late**", with no apparent intervening interval. The Support Engineer can suddenly find that the system design is pretty much fixed, and that it is now too late to influence that design, and because the trials have not begun, we still have no data!

About Us

Aspire Consulting Ltd are a private and independent consultancy who have provided expert Supportability Engineering for over 20 years. Their consultancy, training and software services have been trusted to optimise support systems and validate system design across Defence and Commercial environments.

About the Author

Peter Stuttard is the Chief Executive of Aspire, a Supportability Engineering specialist of over 37 years experience, and an ex REME aviation engineer. He has worked at senior level on a wide range of national and international programmes, from armoured fighting vehicles, submarines and combat aircraft, to tug-boats and communications systems.

www.aspirecl.com

Introduction

The Systems Approach to Support Engineering

As a result of a previous Aspire blog post, **“What is ILS”**, a number of people pointed out that we need to address the design of the equipment, the platform etc. (the “Mission System” in our parlance) when defining (and optimising) support.

Now, in part this was a misconception, because I spent more time illustrating my points by talking about the “Support System” than I did talking about the Mission System, and in part because of the semantics I used. So, for the absence of any doubt, in this article I will use the term “Support Solution” and in this context this includes:

The support aspects of the Mission System, this is the ship, the tank or the aircraft; it is comprised of hardware, software and people.

The Support System is comprised of the entire Support Infrastructure, and the associated support processes and resources.

The Employment Plan defines the manner in which the Mission System will be employed, where, when, in what environment, how often, and by whom. The Support Solution must take cognisance of the Employment Plan and vice versa.

This “Systems” approach is a characteristic of our approach to Support Engineering.

The core problem that I am addressing in this article is a perennial one for Support Engineers; that is that when they try to do the right thing, when they attempt to conduct support analyses, to carry out Through Life Cost predictions, or to optimise the design of the Mission System, for example, particularly so in the earlier stages of a system’s life cycle, they will often be told that it is too soon for such activities. The argument being that the design is not mature and hence there simply is no appropriate data available. Which seems reasonable enough…

Too Soon… Too Late

The problem is that "**Too Soon**" can become "**Too Late**", with no apparent intervening interval. The Support Engineer can suddenly find that the system design is pretty much fixed, and that it is now too late to influence that design, and because the trials have not begun, we still have no data!

The Support Engineering community faces lots problems, both technical and political, but this issue, this demand for data, and the seeming unavailability of that data, is one of the commonest and possibly the most critical. The result is that effective analyses are often not carried out, system modelling does not occur, trade-offs are not conducted, the Mission System is not optimised Through Life Cost [TLC] is not estimated and so on.

Many ILS Managers, LSA Managers and Support Engineers will recognise this scenario.

Now if you cannot estimate TLC you cannot conduct effective investment appraisals and if you cannot conduct investment appraisals how can you evaluate alternative design options or alternative strategies for support? How can you influence the design? How can you do Level of Repair Analyses? How can you prepare a **cost effective** Through Life Management Plan? This problem can sometimes seem impossible, but it can be overcome if the appropriate Systems Thinking and Systems Engineering approaches are applied.

What are these approaches, and how do we define them? The start point is to recognise that many Support Engineering problems are "Wicked Problems" and then we simply need to apply a large dash of logic and common sense.

Wicked Problems?

What do we mean by a Wicked Problem? I stated above that Support Engineers may strive to optimise the Mission System design, this is a classic example of a Wicked Problem, it is highly unlikely that we will ever achieve this, and if we did, we would never know. There are so many variables involved in optimising a complex design, from a support viewpoint, that any proposed solution will make some aspects of the design better, and others worse. Whether the design is better or worse, whether one combination of factors is judged to better than another, will depend on the individual, the organisation and the circumstances in which such an evaluation is being made.

This argument may seem to be a little esoteric but it does have a practical impact; it is important that we understand that there are no absolute answers when dealing with most Support Engineering issues, so we should stop trying to find them. Support Engineers must seek **Relative** improvements, rather than striving to achieve **Absolute** Performance Targets. That is, we must aim for **Better** rather than the **Best.**

Support Engineers deal in statistical probabilities, uncertainties, variability, and uncertain, alternative futures, these are the factors that make Support problems "Wicked", they also affect the nature of the metrics that Support Engineers have to deal with. Typical metrics include Mean Time To Repair, [MTTR], Mean Time Between Failure [MTBF], Failure Rate [λ], Probability of Survival, etc.

Such metrics are determined statistically, they are derived from samples of field or test data, they are not deterministic measures, (e.g. mass). These measures are therefore almost wholly dependent on the availability of some form of historical data.

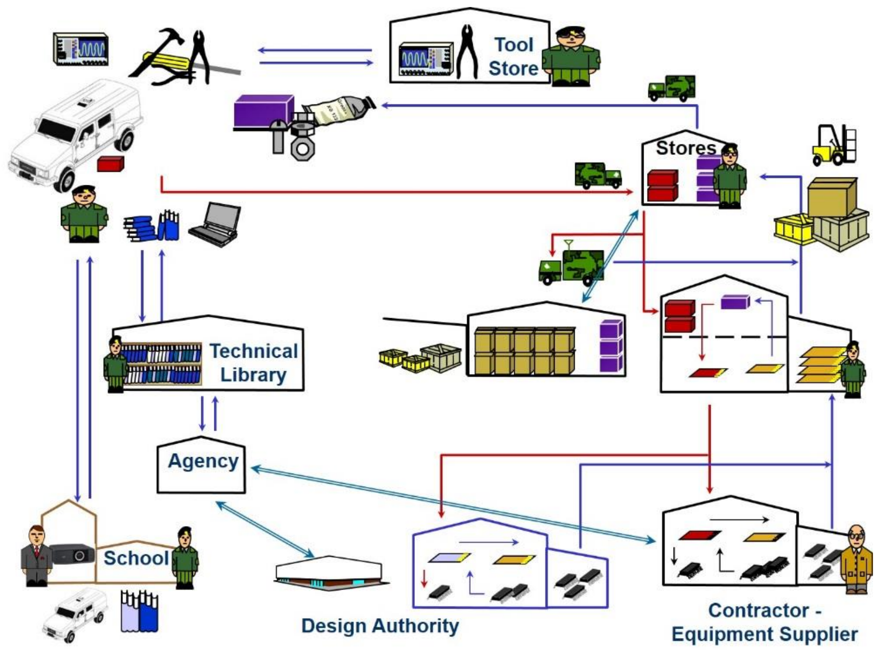
The point of this discussion is that, in most instances we cannot determine the support parameters associated with a Mission System design by analysing the design per se; we must adopt rather more radical approaches…

Do we need data?

This may seem like a dumb question, but do we need quantitative data, for example data associated with our Mission System design? The answer is that this depends on the nature of the particular question that we are attempting to answer.

Consider, if our task were to design a new, improved Support System; let's say for our Land Forces. We would need to capture and to understand the "**design**" of our proposed solution, we would need to understand how it was likely to behave, how it was likely to perform. We would want to compare any proposed Support System "design" with the extant system so we could determine the relative magnitude of any performance improvements, if any.

Now to achieve this we do not need accurate input data, for example precise spares demand rates, what we need is realistic **representative** data. Why? Because in this situation we are testing the 'structure' of our proposed new system so we can determine if that design is better or worse than the existing one; in certain given scenarios.



The Support System – An organisation, a structure, processes and embedded resources

We do not need precise performance metrics if, for example, we want to understand the impact of an improved feedback system, the impact of improving the reverse supply chain, or the ultimate effect of enemy action which is targeting the support elements of a Strike Brigade.

We can model one or more systems, the present system and a future conceptual system or systems, using a common data set for both models, for example common demand rates, common assumptions of attrition (on support resources) due to reliability issues and enemy action, and see how each performs. We are looking to see if there are any "marginal" gains (or losses) and their magnitude, and we will probably be seeking some indication of where further improvements could or should be made.

It is a different situation however if, having finalised the design of our new Support System, we wanted to know precisely how many spare ***Blue Boxes*** we would need to hold, for a given operational scenario; then we would need more precise arising rate data specific to these particular repairables.

Developing Complex System Solutions

Consider now as an example; how would you go about developing or designing an improved Support System for an operationally critical fleet of vehicles for our Land Forces?

Let us assume that this Support System is to handle Engineering Support (excluding ammunition, food and water etc) taking account of consumable items and repairables. Let us also assume that some items will be maintained "In-Theatre", and that our solution must take account of the "Strike" concept, i.e. an offensive column that will penetrate deep into enemy territory. Not all items will be maintained in theatre, some will be returned to a "Strategic Base" either in the UK or perhaps to a "Floating Task Force Base", depending on the location and the nature of the operation.

Now this is a very complex task, how would a sensible person go about this; where would they start, and would they design such a system from scratch?

What they would probably do (what they would **have** to do) is to take the existing system and use it as a start point. They would **have** to do this because the Support System, even for a single fleet of vehicles, is not a stand-alone system. Many elements of any support solution are held in common with a wide range of other Mission Systems. Systems of this level of complexity are not designed from scratch, they evolve over successive generations.

The inevitable conclusion is that we need to identify and define what the extant Support System (the "Present System") looks like. Once this "Baseline" has been captured, we start a structured, rational, informed development process. We identify the characteristics of the "Present System", adopting some of the existing elements that are good and which we would like to retain. We identify the characteristics that are not so good and which need to be improved, we identify those characteristics which present opportunities for improvement, due to the availability of new technologies, new methods, new materials etc.

We can see that we have here the basis of a highly structured development process. What we are doing is applying the principle of evolutionary development, evolving a solution (the "Future System") from a known baseline by applying rigorous systems engineering processes. This is a form of spiral development.

I have used the Support System to illustrate the basic approach, but this concept (that of evolving a design) is as applicable to the Mission System as it is to the Support System.

There is nothing radical about this approach. It is how any complex system is developed; by a process of evolution from previous designs.

The Answer?

The original question was:

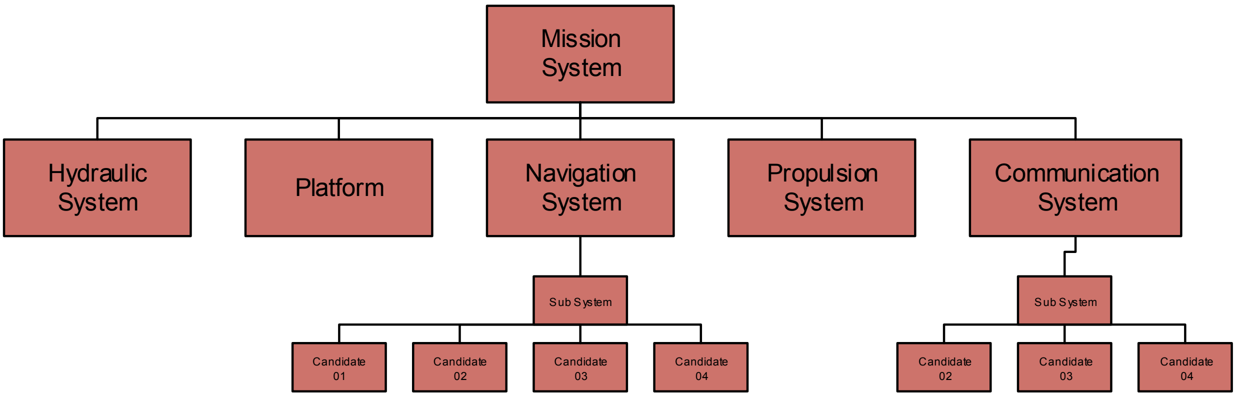
Where do we get the data from to enable effective Support Engineering analyses to be carried out particularly during the early stages of a programme?

The answer is to create a series of baselines derived from one or more existing systems. Now, because the Support Solution is comprised of the Mission System, the Support System and the Employment Plan, then we need to establish three separate baselines.

This approach is not radical; the original standard (MIL STD 1388) for Logistic Support Analysis [LSA] calls for the development of a Baseline Comparison System. The dreaded Logistic Support Analysis Record [LSAR] incorporated a facility (albeit a clumsy and flawed one) for capturing such Baselines. Similarly, those of you who are familiar with LSA standards will know that the "Use Study" requires us to determine how the proposed system will be operated, and it also requires us to document the existing support arrangements (i.e. to document the Employment Plan and the Support System). What these standards failed to do was to define how this information was to be captured and how it was to be managed once it had been captured.

If we create these three baselines we will have the data that we need; or at least we will have the data that we need to "Prime the Pumps". If we do this we know what is possible, the minimum that is achievable, for the simple reason that it already has been achieved; we then begin the process of evolution.

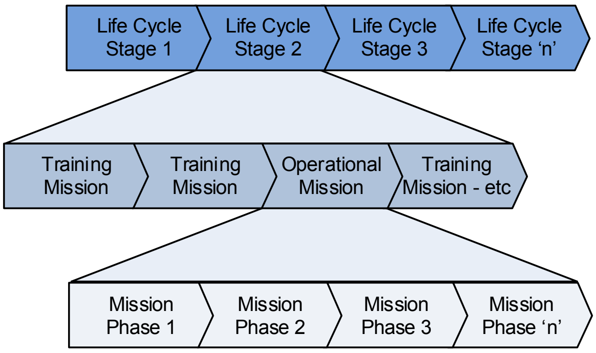
Such baselines need to be structured, meaning we need some mechanisms and some tools for capturing the information. There are many options for capturing a Mission System definition and all the associated characteristics that make such a system supportable or not. These options include the LSAR, but there are also a wide range of other choices, including Product Data Management [PDM], Product Lifecycle Management [PLM], CAD, CAE tools etc.



Tools for capturing and managing Mission System data tend to be organised hierarchically

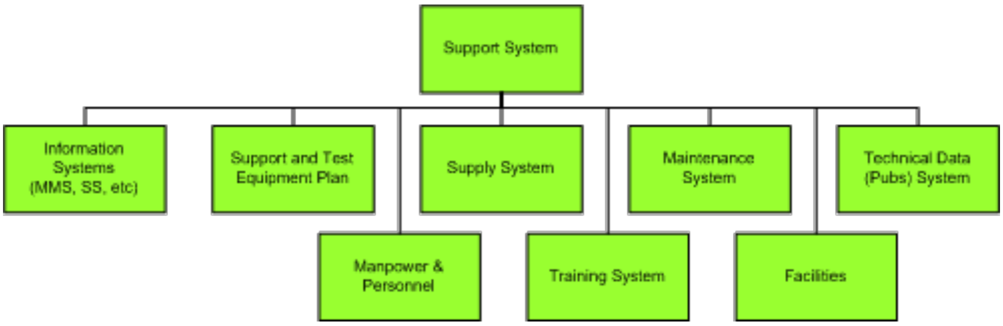
The same is not true for the Support System or the Employment Plan. There are no 'off the shelf' tools; here we tend to find the default tool is the basic word processing software we all know!

But Employment Plans can be described in terms of Life Profiles, these in their turn can be decomposed into Missions (Training and Operations), Missions can be broken down into Mission Phases, etc. That is, there is a simple form of hierarchy here also.



Employment Plans can be represented in the form of a simple hierarchy

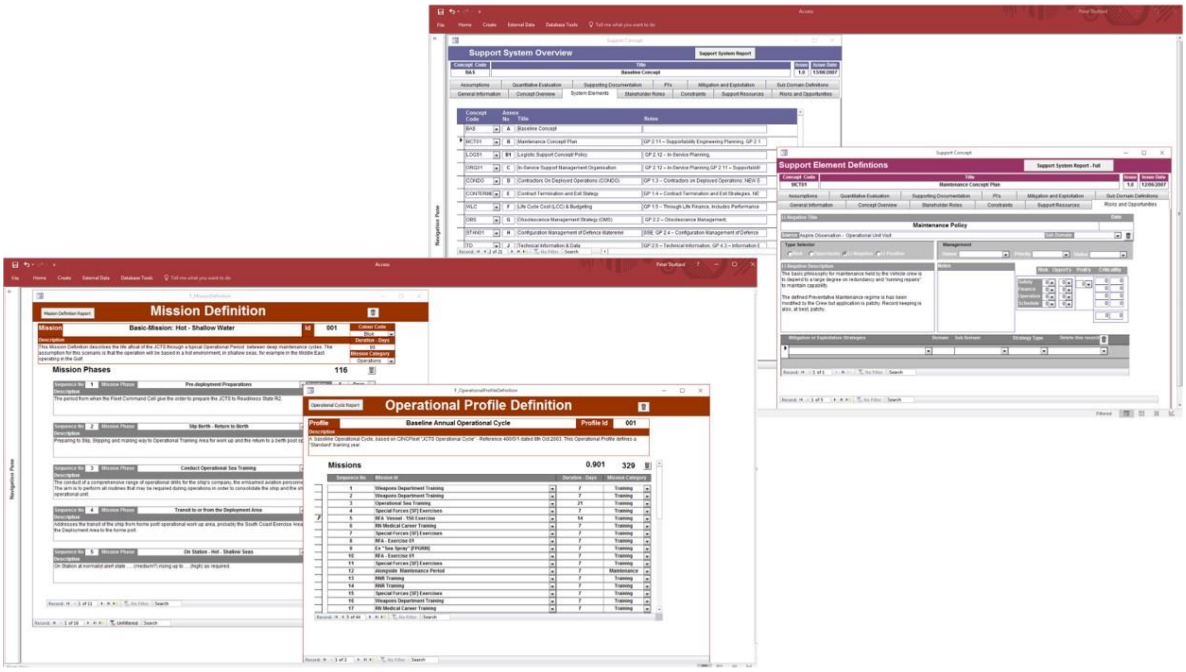
Similarly, the Support System can be decomposed into the key support elements. For example; Supply Support, Training Provision, Technical Publications Management, Maintenance Planning etc.



The Support System defined in the form of a simple hierarchy

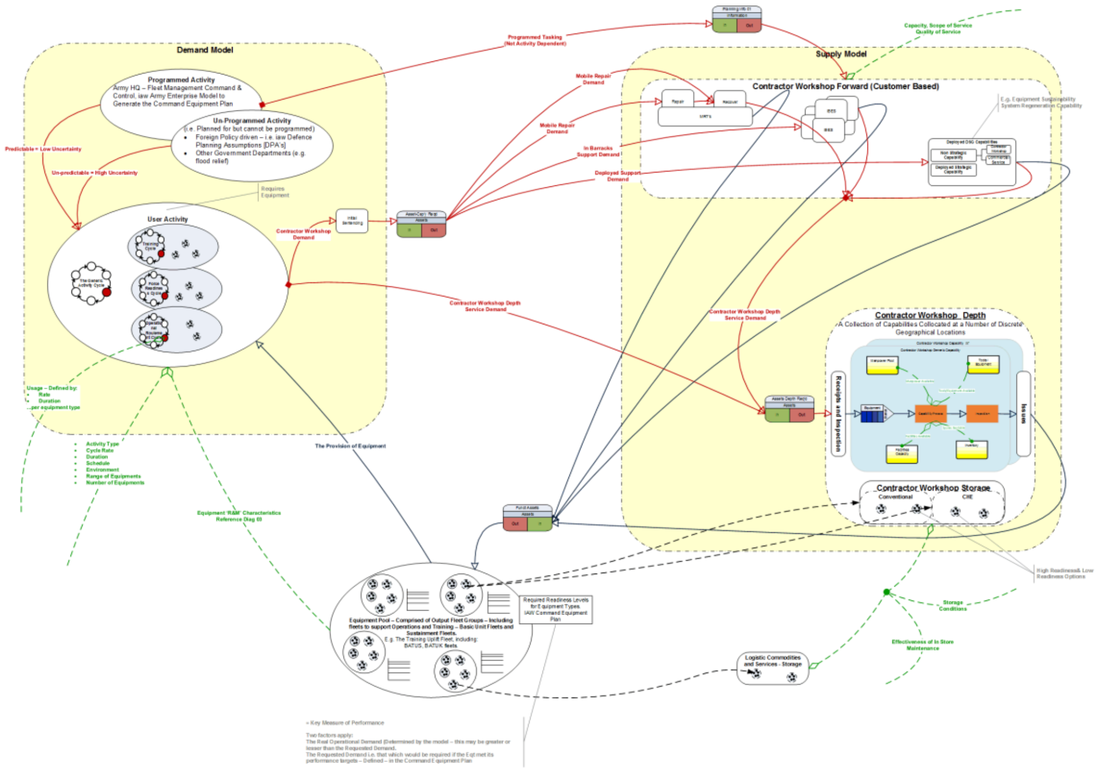
Such hierarchical approaches are amenable to the use of databases, and the use of hierarchies in such databases means that we can use these structures to 'hang' all sorts of data and information on, information such as:

- **Lessons Identified** – Both positive and negative – a deceptively simple but very powerful tool in our armoury – these are a backward-looking variant of…
- **Risks and Opportunities** – which are forward looking.
- **Interfaces** – which apply to all three elements. Consider the Support System; we can identify the interfaces between the Customer and the Contractor, these may be the points at which physical items transfer from one organisation to the other. Interfaces are where information is transferred, or even where responsibility or ownership is transferred. This is important because it is to such interfaces that we need to apply appropriate…
- **Performance Indicators (PIs and KPIs)** – which will form the basis of Contracts and Service Level Agreements (SLAs), and…
- **Assumptions**, **constraints**, etc.



Aspire's tools for capturing Employment Plans and Support Systems – the Employment Plan Definition Environment [EPDE] and the Support System Definition Environment [SSDE]

In all three cases, Mission System, Support System and Employment Plan, such databases can be (should be) supplemented with other techniques, such as process flow diagrams, rich pictures, influence diagrams and so on.



Graphics, rich pictures, process flow, influence diagrams etc are needed to complement definitions captured in a database.

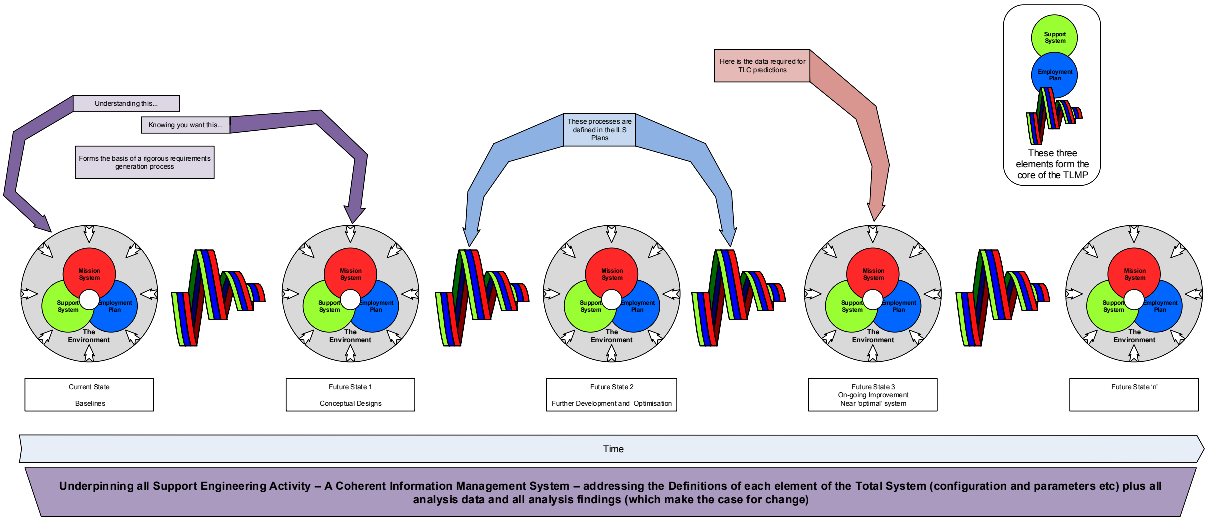
These baselines are now subject to a continuous process of evolution. For example, design improvements should be identified that will address negative Lessons Identified, to manage Risks or to exploit Opportunities.

We can 'Test' alternative Mission System concepts in the context of the evolving Employment Plan and the Support System concepts, alternative Employment and Support Concepts can be documented and evaluated. The databases and any diagrams are updated as the "Support Solution" evolves, as we identify and define alternative concepts and as we make 'design' decisions.

Once the log jam is broken…

The benefits of this approach are profound, consider:

- The data from initial Baselines can be used to populate Through Life Cost [TLC] and Availability models, thus providing a "Benchmark" against which any improvements can be compared.
- Similarly, at any stage, as the baselines evolve into "Concepts" the new definitions, the databases and the schematics, can be used as a source of data to populate not only TLC models, but also Availability, Level of Repair Analysis (LoRA), Sparing and Manpower models; enabling us to quantify progress.
- This method of utilising Baselines solves a critical problem for Support in the Defence sector; namely the setting of SMART requirements. These Baselines provide the foundation for a robust, structured, logical requirement generation process.
- When we define the processes required to evolve from one state to the next we are creating a programme plan. This delivers a Through Life Management Plan (TLMP) by default; comprised of the Support System Definition, the Employment Plan Definition and the Programme Plan.



Many other benefits accrue as a result of this approach…

The Cost

Creating and then evolving such baselines does require investment, but it is insignificant when compared with what is traditionally spent (and certainly when compared to the benefits of such an approach). The approach is likely to pay for itself in terms of improved programme efficiency alone.

The first attempts will of course be the most challenging, but consider the amount of overlap and duplication that exists between different programmes. Consider the similarities between two Royal Fleet Auxillary (RFA) vessels, or even those between an RFA and a Royal Navy (RN) vessel. The degree of commonality can be significant. Hence much of the work could not only be reused, but such an approach would also facilitate much needed coherence across a range of systems and their support arrangements.

There will be a great deal of commonality across projects …

Conclusions

The Digital Twin Concept

It is time we seriously considered concepts such as the "Digital Twin"; a simulated companion of the physical asset, representing their (near) real time state. This applies not only to assets such as gas turbine engines (where the digital twin concept is becoming common use) but to all platforms: wheeled and tracked vehicles; ships; aircraft; and their associated Support Systems and Employment Plans. Such an approach would deliver significant benefits, not only during the development of a new system, but also during the management of extant In-Service systems. Today's technologies render this feasible with relatively low levels of investment.

However, this does beg a question; which is "how good is the historical data ?" There is no doubt that a great many in-service data collection systems do need to be improved, along with the quality of the data they collect. This is more likely to happen if (or when) that data is effectively exploited.

In Summary

The creation of baselines and the evolving of those baselines via a series of stages, until a formal Support Solution is achieved, will address the issue of the apparent lack of available data. The approach is simple and readily achievable, and it will deliver significant benefits, beyond addressing the data availability issue alone.

We are dealing with "Wicked Problems" hence we seek **relative** improvements rather that **absolute** performance targets.