14

Setting the Systems Methodology to Work

I like work: it fascinates me. I can sit and look at it for hours. I love to keep it by me: the idea of getting rid of it nearly breaks my heart.

Jerome K. Jerome, 1859–1927

Previous chapters have presented the systems methodology in abstract form: whereas the tasks to be undertaken have been nominated and interrelated, there has been no mention of the manner in which the undertaking of the many and various tasks involved might be established, organized and set to work, to achieve the goal — the solving of some problem. This chapter examines how the systems methodology as a whole may be 'overlaid' on to, or mapped into, real world organizations and structures.

Systems Methodology in Phases

There is a natural, inalienable sequence embedded within the systems methodology: essentially, some things have to be done before others, and some things can only be done based on what has gone before (see The Time dimension on page 167). It is possible, then, to consider the systems methodology as comprising a number of contiguous phases, such that the output/outflow from one phase becomes the input/inflow to its successor. So, there might be:

1. A problem-solving phase that addresses the issues, explores the problem space, and emerges with one or more conceptual remedial solutions.
2. A phase of examining how conceptual remedy might work, how it might go about achieving its objectives, how achieving those objectives might be threatened in the real world, and what might be done to neutralize the threats.

Systems Engineering: A 21st Century Systems Methodology Derek K. Hitchins

ISBN: 978-0-470-0585-5

3. Following on, a phase to identify a range of functions that the putative solution system should be able to perform, both to achieve its objectives of solving the original problem, and of sustaining itself as a viable system.
4. The many and various functions can then be related and interrelated so as to develop an intrinsic functional architecture, which can be mapped into physical partitions to accommodate dynamic solution space constraints of space, economy, and time — the optimum design.
5. In the light of this optimum, ideal world solution system design, a phase may follow in which choices are made as to whether to solve, resolve or dissolve the original problem — so, potentially, changing the direction and outcome within the applied systems methodology . . . a solution system will be required to resolve the problem, and a solution system — perhaps of a different nature — may be needed to dissolve the problem.
6. According to the choices made in 5, there may follow a phase of specifying, developing and making physical parts, while at the same time recruiting/acquiring and, if necessary, training people as members of human activity system (HAS) teams.
7. The various complementary parts forming the whole SoS may then be interconnected and 'encouraged' to work together — this might be termed a work-up phase, during which the performance of the whole is progressively improved until the whole exhibits requisite emergent properties, capabilities and behaviors
8. The SoS will then enter its 'operational phase,' during which it will continue to solve/resolve/dissolve the problem for which it was created, and will continually repair and redesign itself as the problem evolves/morphs.
9. Should the problem for which the SoS was created cease to exist, the SoS may then be 'phased out,' or, alternatively may be 'repurposed.'

There may be many variations on the theme expressed above: solution systems may be introduced piece by piece, perhaps to aid affordability or to minimize organizational turbulence; and so on. However, one way to organize the implementation of the systems methodology is simply to form a team of suitable individuals, to train them to work as a team, equip them with the necessary tools to address each phase in turn, and encourage them to practice. . . .

Systems Methodology as HASs

There can be many ways to arrange and organize the systems methodology for practical application. Figure 14.1 illustrates the point; while there may be a 'natural sequence,' it does not follow that the organization need mirror that natural sequence precisely, although it is likely to follow it in some degree. Moreover, not all situations require the full extent of the systems methodology to be 'applied;' sometimes it will be necessary only to address the problem and produce remedial solution concepts, for instance.

On other occasions, a group of people may be presented with a solution concept, already selected by some customer or arbiter; it may be the responsibility of the group to work from that point onwards, without investigating the remedial solution options. (Such a group may elect, judiciously, to investigate the problem and the options anyway, so that they understand the nature of the problem they are being asked to solve. . . .)

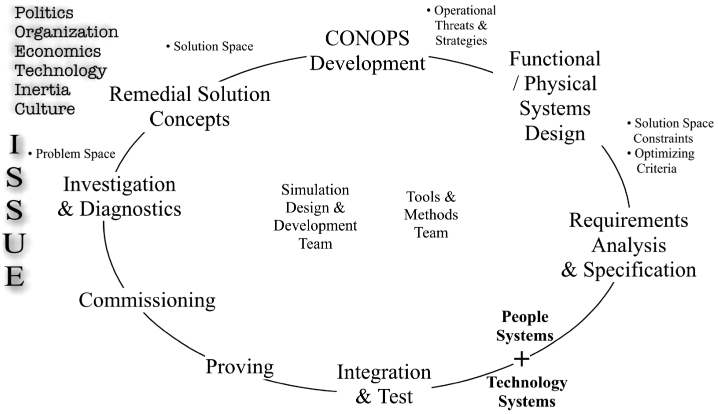

Figure 14.1 Systems methodology 'in the round,' showing principal activity centers, which may be manifested as individuals or as teams of people forming human activity systems (HASs). So, Investigation and Diagnostics could be the team title for a HAS formed of analysts with knowledge of the particular domain(s) of interest. Similarly, CONOPS Development might be the title of a team concerned with elaborating and testing various concepts for the operation of the solution system with a view to selecting and validating the most appropriate CONOPS to solve the problem in the circumstances . . . .

Systems Methodology as Tools

An alternative view of the systems methodology illustrates it as a pipeline of successive processes, many of which may benefit from the use of context-free systems tools, methods and techniques. Such tools enable systems methodological practitioners (systems engineers?) to handle and accommodate amounts of information that might otherwise swamp them, or at least prevent them from seeing the forest for the trees.

Figure 14.2 shows a tool-rich view of the systems methodology, using a selection of tools that have proved useful; there may be many others. The figure starts top left with an Issue and a Problem Space, which is addressed in the example using the Rigorous Soft Method, which may provide one or more conceptual remedial solutions — always supposing there to be any solution. The systems methodology then proceeds as illustrated, following the arrows at top and bottom of the diagram. In the center, the Generic Reference Model and the Automated ISM (Interpretive Structural Modeling)/N2 Chart are shown as being employed in several activities/processes.

Similarly, the use of a proprietary dynamic simulation tool, such as STELLA™ is indicated; there are several others. However, it should be noted that such tools are not used as their manufacturers might have expected; in particular, within the systems methodology, the dynamic simulation tool simulates whole, open, adaptive, interactive systems rather than phenomena, and the same tools are used to undertake optimization through cumulative selection — for which task, future tools yet to be developed may be better suited.

Finally, at bottom right, the solution system emerges — in full design, at least, since this is as far as the figure progresses. Note that information from the problem space — dashed line — is passed along the process line to prove the solution system design, which must be able to solve the original problem.

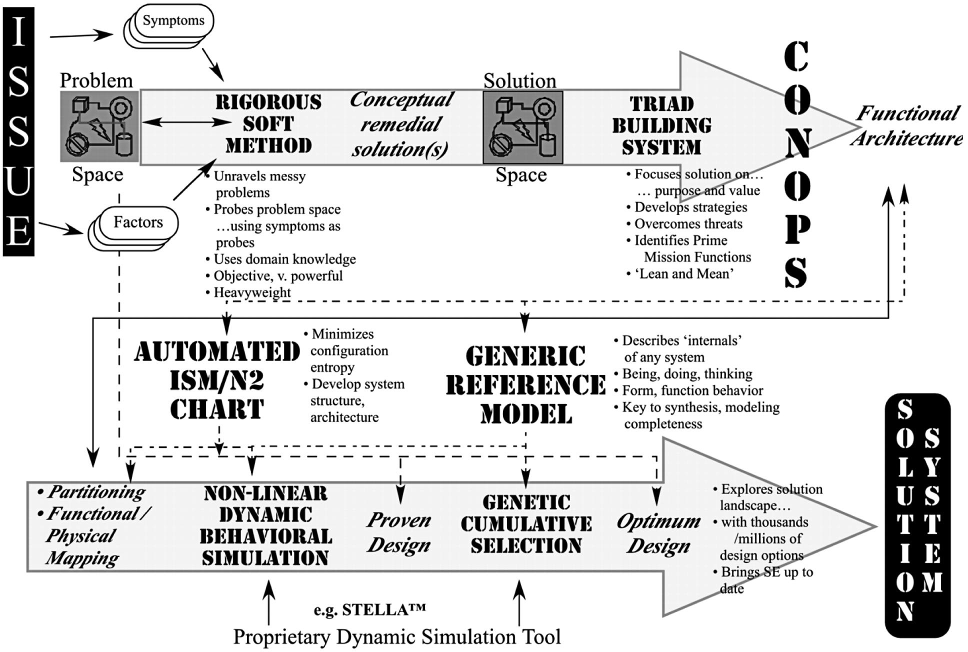

Figure 14.2 A typical assemblage of tools and their points application within the progress of the systems methodology. The tools are all context-independent, but have in common their facility for reducing information and configuration entropy, leading from a disordered, perhaps even chaotic, issue, to an ordered, specified solution system design.

Organization for Applying the Systems Methodology and for Systems Engineering

There are many ways in which an organization for the implementation and application of the systems methodology may be set up and managed. In the past, and particularly in the 1960s and 1970s, such organizations were commonplace in the so-called systems houses — these were independent companies who explored problems and issues and created solutions to those problems for customers. Systems houses treasured their integrity and objectivity, and owed allegiance to no company or brand name that might predispose them to use facilities or equipments that were less than appropriate. So, they rarely manufactured — instead they either selected and bought-in suitable equipments, or put build specifications out for tender to engineering companies to make what was needed.

Figure 14.3 shows one such, in which the organization was managed under five main headings, as shown. It is noteworthy that the objective of operations and requirements analysis, some 40 years ago, was to find the 'real' operational requirement.

This was because it was common practice for customers to believe that the system they wished to procure was to solve one problem (expressed in a customer's so-called operational requirement), but for the systems analysts and systems engineers to find, upon proper investigation, that it should

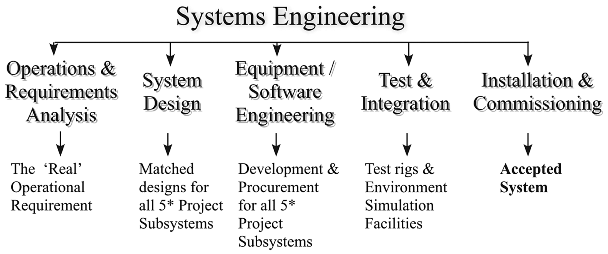

Figure 14.3 Systems engineering organization, *circa* 1965–1975 (Hitchins, 1992). The systems methodology 'process' proceeds from left to right. See text for the '5* systems.'

have been to solve quite another, and in an unanticipated way. Operations and requirements analysis undertook the task of exploring the problem space, of finding conceptual solution remedies and then of identifying the putative solution system and its CONOPS, usually by extensive simulation of competing solution concepts in their future operating environment. It was not uncommon for the systems engineering team to come up with a solution to a problem that was quite unexpected by a customer — often one that not only solved the problem effectively, but also cost considerable less.

System design, as illustrated, produced matched designs for five project subsystems:

1. The primary/operational system — the system that the customer expected.
2. The in-service maintenance system that the customer would require to keep the primary system operational.
3. The in-service training systems needed to train maintainers and operators, including crew simulators, part-task trainers, maintenance and servicing rigs, etc.
4. The in-company engineering system needed to develop 1–3.
5. The in-company maintenance system needed to keep the engineering system operating.

There were, and are, relationships between the various systems. For an aircraft or ship project, for instance, a crew simulator would have the look, feel and behavior of the corresponding operational aircraft or ship, so that the crew felt that the training was entirely realistic. During test and integration, there was also a need for simulation, such that the displays and controls, for instance could be tested and their behavior realistically assessed. There was evidently a relationship between the two kinds of simulation.

Similarly, there was a relationship between the facilities needed to maintain the equipments that had been bought-in or manufactured, and were to be integrated, and the same equipments when they were in service. For instance, in-company test and maintenance might involve factory test equipment (FTE), while in-service maintenance might involve automatic test equipment (ATE); although different in some respects, there was a degree of commonality that would have made it sensible to develop ATE from its precursor FTE.

So, as long ago as the mid 1960s, if not earlier, the practice of creating whole solution systems, where the operational, or mission element was only a part, was in evidence. It happened only rarely, however, because procurement agencies of the time were determined to maintain competition and to spread aerospace and defense contracts around, regardless of overall cost or integrity of the whole solution system.

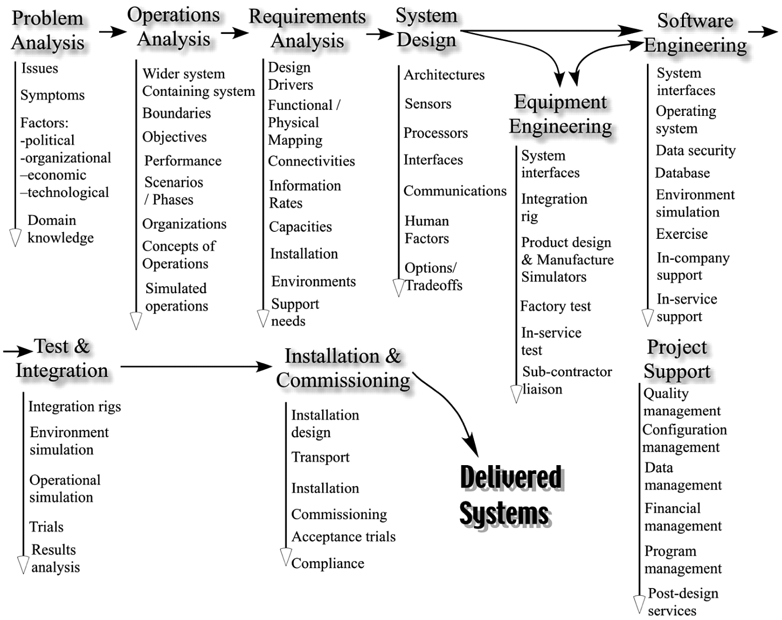

Figure 14.4 Systems house project organization, showing tasks, skills and activities; *circa* 1965–1975 (after Hitchins, 1992).

Not evident from Figure 14.4 is the systems methodology — the 'how' of systems engineering that runs as a weave across the teams' vertical weft. So, in addition to such figures about organization, there has to be some process/project/program plans and charts which show what activities are to be done in what sequence, what inputs such activities will need, what tools may be used, where the output from the activities is expected to go, and so on. This chart effectively runs laterally across that of Figure 14.4, so that skills, capabilities, etc., that are required to undertake tasks are seen as being available as required.

SM GANTT charts

Figure 14.5 shows a notional GANTT chart which might be used for planning an application of the systems methodology. Time progresses along the x-axis. The burden of work passes progressively through the various teams: Problem Analysis, Operations and Requirements Analysis, Systems Design, and so on. Although work passes from team to team as they address the various systems methodological phases, it does not follow that the teams are comprised of different practitioners; on the contrary, it is likely that some practitioners will move with the burden of work, from team to team, according to their skills, expertise, domain knowledge, etc. The resources section at the bottom could be filled in for a particular project, enabling the beginnings of project cost assessment. Note also in the chart that milestones — often marking major deliverables — will occur towards the end of each phase. Program evaluation and review technique (PERT) Charts can be employed in similar fashion to establish detailed patterns of work.

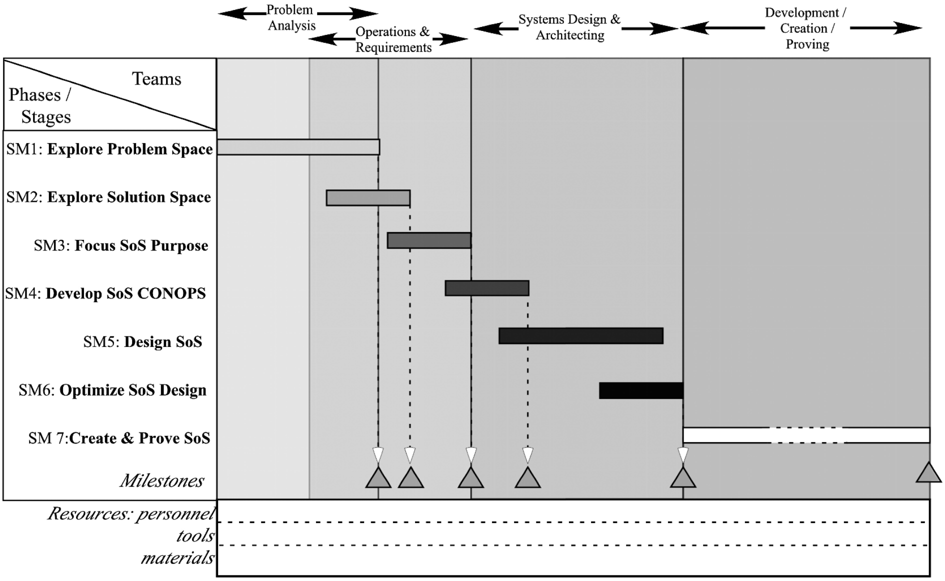

Figure 14.5 Notional GANTT-type planning chart for an application of the systems methodology.

Teams of teams

For large and/or complex projects, a team of teams may be appropriate – see Figure 14.6. Such an arrangement would not be uncommon for projects such as Apollo, in the past, and for major civil engineering, aerospace and defense project today. There may be top level team, which addresses the issues and the problem at the top level, generates conceptual remedial solutions, develops a number of potential CONOPS — perhaps several for each remedial solution concept — and progresses towards a putative set of open, interacting, adaptive, complementary viable subsystems. Each of these subsystems may be a complex system in its own right, and so properly merits its own systems team, applying the systems methodology. Indeed, the resultant arrangement would be a team of teams. . . .

However, the teams cannot sensibly work independently, else the various subsystems may not remain complementary. At the top level, each of the subsystems will be prescribed in terms of its dynamic emergent properties, capabilities and behaviors, its interactions and interfaces, etc., or to use the apposite engineering term, 'fit, form and function.' Also, the various subsystems must be compatible with the CONOPS of the whole, such that when the various parts are brought together, they contribute severally and together to the prosecution of the CONOPS.

In practice, a high degree of harmony may be required between all the teams and this is may be achieved by including members of first-level teams in the corresponding top-level team, and *vice versa* — remembering, of course, that teams of more than half a dozen find consensus and decisions increasingly hard to reach. The development of CONOPS, the functional design, the functional/physical design, etc., may all be the subject of continual variation, as concepts and designs of parts impinge on the whole and as the constraints of the whole impinge on the parts. For example, were the emerging design and development of one of the subsystems to vary significantly

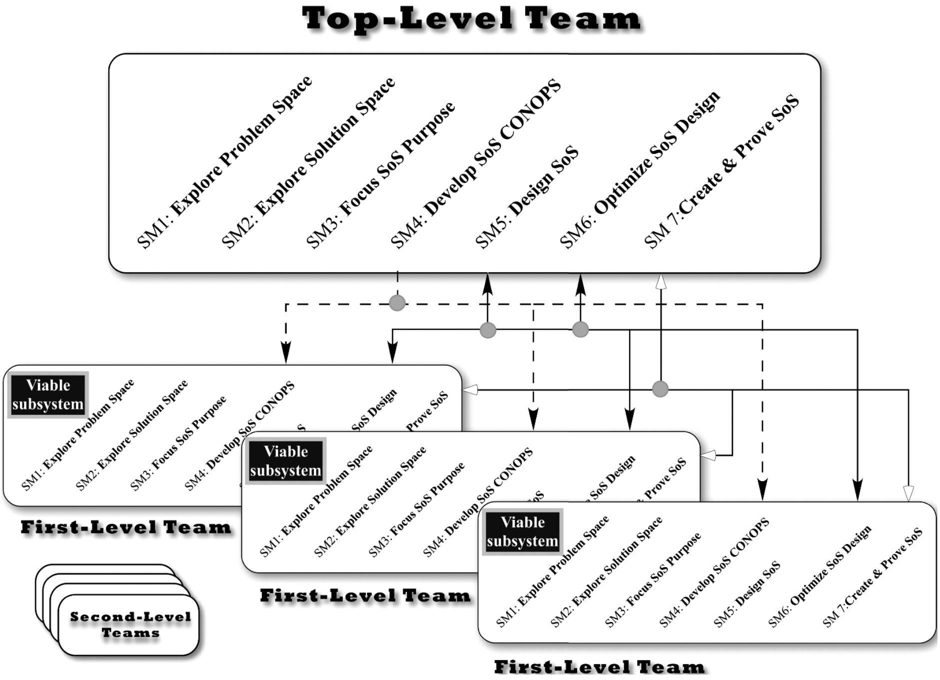

Figure 14.6 Systems methodology as a team of teams.

from expectations, it might be necessary to redesign or reconfigure the offending subsystem; or, if the situation could not be ameliorated, then other subsystem may have to be 'adjusted' to accommodate the imbalance, resulting in an all round redesign.

An early indicator of impending difficulties may be found by budgeting; that is, by periodically assessing and evaluating key parameters. For instance, in a largely technological solution system, it may be prudent to check forecasts against budgets for mass, volume, centre of mass, shape, moment of inertia, capacity, endurance, energy consumption, dissipation, waste, processing power, communications rate, etc., etc; i.e., many of the various emergent properties, capabilities and behaviors of the subsystems, which contribute to the similar emergent properties, capabilities and behaviors of the whole.

Team of teams and inner/outer loops

Figure 14.6 presents an alternative view of outer loop – inner loop design as previously seen on page 293; in the figure, the perspective is one of hierarchical, rather than 'inner loop contained within outer loop.' A moment's consideration, however, will convince that they are really the same — either viewpoint is rationally tenable; each equates to the other. The figure shows a top level and several first-level teams; there could be further levels, effectively forming the downward extending roots of a tree of systems teams.

Team of teams and system of systems

As we have seen, the developing design of a complex system may support the formulation of a number of complementary subsystems: these may be functionally bound subsystems, or they may be viable subsystems, i.e., systems which may stand on their own in their environment.

The difference between functionally bound subsystems and viable subsystems may be relatively trivial, or it may be significant. Functionally bound systems may exist within a whole that has perhaps complex, but nonetheless unified, resource management: similarly, viability management may be complex, but singular. Example of such organizations might be the human body, with its many, tightly functionally bound subsystems, all of which are resourced via the one complex system which provides energy, removes waste, replaces aging cells, etc. Similarly, for the body, the management of viability is undertaken for the system as a whole (the whole body) with the organs being addressed as composite parts of the whole, rather than as separate subsystems. An alternate example of this arrangement might be an enterprise operating from one location, with various departments and sections, and with resources and viability being managed locally for the whole.

There can be many arrangements, according to the design outcome, with some subsystems tightly functionally bound, while others may be viable; in some instances, all of the subsystems may be viable. This last may come about in a different manner, with the formation of a whole by the bringing together of a variety of extant operating systems under one ‘umbrella’ to form an association of systems. Such an association need not be a system — it may not satisfy the simplest definition of the term ‘system.’ However, it may be capable of becoming a system, perhaps post-association, with the various parts and their interactions being configured and adjusted to make the parts complementary, cooperative and coordinated, usually under the guiding hand of a directing, controlling (orchestrating?) entity.

Figure 14.7 shows how a number of subsystems within an overarching containing system may be progressively conceived, designed and developed, where the subsystem emerging from the design process are viable systems. Examples of such system are legion, and include Apollo, where the mission subsystem was comprised of a number of craft (command module, lunar excursion module, etc.) that, although fitting together like a 3D jigsaw puzzle, were also capable of independent operation as viable systems in their own right. A second example might be of an aircraft fitted with short-range air-to-air missiles for self-defense. The whole aircraft, with its crew, is a viable system. The missile, once launched, is a viable system too, with its own purpose and control, its own sensors, effectors, resources (propellant, refrigerant, payload. . .), etc.

As the figure suggests, there may be several teams operating, perhaps at the same time, perhaps not, but all would be operating within the one systems methodology. The top-level team is concerned with solving the root problem, and in pursuing the systems methodological process; it will establish a preliminary overall design, which may include a number of potentially viable subsystems. The top-level team may then spawn, or otherwise encourage the formation of, first-level systems teams, and the systems methodological process may then be repeated, but with all the teams working in coordination. The CONOPS for the whole system need not be the CONOPS for each viable subsystem part, but the conceivers and designers of those parts need to be aware of, and understand, the overall CONOPS, so that the CONOPS for the parts both enable and contribute sensibly to, that of the whole.

Similarly, the purpose, behavior and mission features of the parts support, enable and contribute to the purpose, behavior and mission management features of the whole. As Figure 14.7 shows, the mission management features of the whole will plan (choreograph) and coordinate (orchestrate) the various subsystem activities such that their combined functional behavior results in the requisite dynamic emergent properties, capabilities and behaviors (DEPCABs) of the whole.

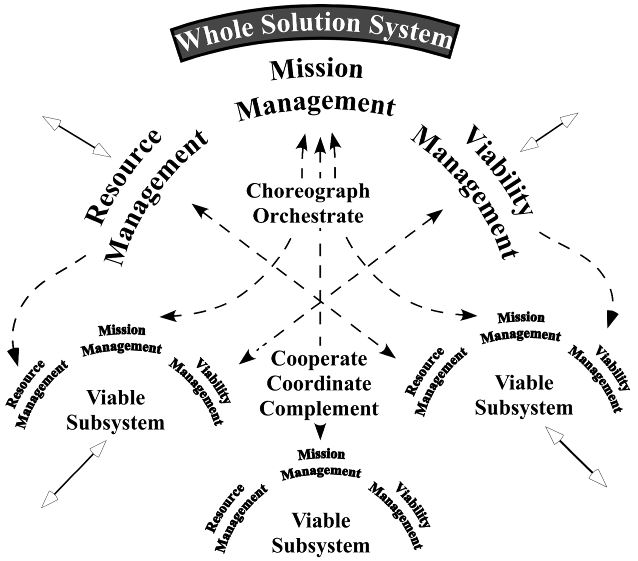

Figure 14.7 The whole solution system manifests resource, viability and mission management. Mission management for the whole choreographs (plans) and orchestrates (activates/coordinates) the various subsystems, each with its own discrete management capabilities, to coordinate the execution of missions. The whole will be neither fully resourced, nor viable, unless its subsystems parts are both resourced and viable.

However, since some of the subsystems may be viable systems in their own right, resource management for each viable subsystem may be independent — at least, in some degree. At a fundamental level, resources come in two varieties: throughput resources, that the system takes in, processes, and passes on; and resources to maintain, replace and build up the internal features of the system, to establish, maintain and improve its capabilities. Provided that the DEPCABS of the viable subsystem remain as required, this second variety of resources may be independent of such similar resources for other viable systems in the complementary set making up the whole.

An example will make the point. A global lean volume supply system may be comprised of many parts, major viable subsystems in their own right, formed into a fan-in pipeline of subsystems with the last ‘pipe’ in the line being final assembly, output, marketing and selling to the consumer. The whole system is choreographed and orchestrated by this lead system, the one that interface with the market. However, some of the various plants in the pipeline may be in different countries, may employ people of different nationalities and cultures, and may employ different machinery. Resources that maintain, build up and replace these parts do not have to be the same for each of the various viable subsystems, provided that the performance and behavior of the whole is reflected in that of the parts. So, while the throughput and quality of the manufactured and assembled parts is choreographed and orchestrated from the center, the people and machinery used to create and process those throughput resources need not be, provided their behavior is as prescribed and expected.

Similar considerations apply to viability management (S-MESH); provided the viable subsystem remains viable, the means by which it achieves this end may be of less concern in some respects. Synergy is the cooperation and coordination between the various parts to produce some desired external effect — the same (emergent) effect may be produced using different parts (people and machinery in this case). Maintenance similarly implicates chiefly 'local' people and machinery. Evolution may arise locally, but may require direction from the containing system center, such that the evolution of the parts contributes sensibly to, and is compatible with, evolution of the whole. Survivability of the viable subsystem part may be a local consideration, where there are distinct local threats, but it may also be a whole system consideration where, for example, interactions between the parts are threatened. Homeostasis, as ever, is a sensitive consideration. Viability for the viable part requires homeostasis, notable of local resources and capabilities. Viability for the whole, on the other hand, requires a continuing homeostatic balance between inflows and outflows, not only for each viable part, but also for the whole.

Returning to Figure 14.7, it can be seen that the various systems design teams at top level and first level may apply the same systems methodology, and may progress the top level design hand-in-hand. At the same time, the first level design teams will be identifying and defining the implication of the top level design team's findings as they 'flesh out' the specific CONOPS, mission management, resource management and viability management features of their respective subsystems, such that the DEPCABs, rather than internal substance, of viable subsystem parts, is realized.

Summary

Setting the systems methodology to work may be as simple in concept as bringing together, training equipping and practicing a team of suitably qualified and experienced people, but there are nonetheless a variety of viewpoints on the subject. The systems methodological process may be perceived as a series of contiguous phases, for example, with the output from each phase forming the input to the following phase. An alternative, but related view sees the process as punctuated by the use of different tools, each appropriate to one or more phases, such that the information that emerges from one phase and its tool/method/technique is in the right form and format to serve as input to the next phase, tool/method/technique, making the whole systems methodological process 'seamless.'

There has been, in the past, a variety of successful way of organizing the systems methodology process (the 'how' of classic systems engineering) such that it is conducted by a number of teams within a systems engineering organization. Examples are given of such arrangements from the 1960s/70s, as employed by so-called systems houses, which were objective, independent systems engineering companies set up to employ the systems approach to solving complex problems in the defense, aerospace and civil engineering arenas, including hydroelectric energy, atomic and nuclear energy, and many more.

These successful examples from the past serve as a guide for organizational paradigms of the present and future. The chapter continues with a look at modern organizational paradigms, with the employment of teams and teams of teams, applying the systems methodology to systems and systems of systems.

Assignments

1. You are Vice President, Corporate Organization, of a manufacturing organization that has recently acquired five major companies: two for manufacturing parts, one for storing and transporting parts, one for marketing, and one for distribution. Each of the acquired organizations

was a viable, profitable business in its own right prior to acquisition. The President of your organization sees the future as being largely concerned with agile, lean volume supply systems, and it is with this goal in mind that he has acquired the new subsidiaries. He is aware, however, that he does not yet preside over a system of systems. Indeed, he feels unable to regard them as even an association of systems, since each of the acquired subsidiaries remains fiercely independent under its own operational control. The President goes so far as to call the subsidiaries 'a dissociation of systems!'

- The President has decided to set up a working party, with members from each of the five subsidiaries. The terms of reference for the working party are, broadly, to consider how the various subsidiaries may be 'harmonized' (his term) into a unified, lean volume supply system.
- You are to head up the working party, but in the first instance, the President requires you to determine the agenda for the working party: you have one day.
- You retire to consider what needs to be addressed, discussed, agreed, planned and executed in order to reconfigure the separate subsidiaries into an agile, lean volume supply system . . . after nervously scanning the recruitment advertisements in the trade magazines, you settle to work and produce the draft agenda. . . .

2. You are a member of the top-level systems team addressing the problem of sending a team of astronauts to land on Ceres, one of the largest of the asteroids orbiting the Sun. There is already a team of system teams in existence, with first-level teams for ground environment, communications, imaging and graphics, and for the various parts of the mission system — launcher, command modules, etc. Quite how physical contact with the asteroid is to be accomplished is still under discussion as part of the top level CONOPS. . . .

   - You, however, are tasked with examining a different subject: software. It has become the practice to form a central, software development center, with responsibility for developing all software, including that for each of the discrete parts of the mission system. This centralized approach was introduced many years ago within your organization to standardize and improve the quality and reliability of software
   - This centralized approach has created some difficulties in the past, since each of the mission subsystems, with its respective first- or second-level systems engineering team, has no direct responsibility for its own mission software. There is a proposal to rectify the situation by 'embedding' the software for each viable mission subsystem within that system.
   - So, the systems engineering team for the command module, for instance, would be directly responsible for conceiving, developing, testing and proving the software, just as they would any other piece of the command module.
   - Remembering that the CERES Mission Program already operates as a systems engineering team of systems engineering teams, your task is to consider the pros and cons of centralized versus distributed/embedded software development in this context and to present a considered recommendation — with justification — as to which way to organize software development, integration, test and proving.