12

SM7: Create and Prove Solution System (SoS)

Prove all things; hold fast that which is good.

Bible, Thessalonians

Introduction

Previous chapters have developed a functional/physical design for a solution system, with purpose, functions, behaviors, architecture, configuration, etc. This chapter sees the systems methodology continuing towards tangible creation and operation — supposing that, for the time being, to be the requisite outcome.

The general approach is outlined in the behavior diagram of Figure 12.1, and commences by confirming the system and subsystem design specifications, which were an output from prior activities. The systems methodology develops a test environment, perhaps an open, dynamic, interactive simulation model, which represents other systems with which the SoS will be interacting, plus context, environment, scenarios, etc. While in some cases this simulation might be a complex, real-time computer-based virtual reality, in others it could be a number of people, perhaps in teams, adopting the role of other (people) systems: in yet other cases, it may be possible to interact with real world systems. Importantly, the test environment, simulated or real, must be representative, open, adaptive and reactive in real time; else, it will not serve to test the design effectively.

The simulated SoS, with its subsystems and interactions, is brought together with the dynamic test environment. As the various subsystems interact, they should exhibit the full range of dynamic emergent properties, capabilities and behaviors (DEPCABs) anticipated in the design, as the whole SoS interacts dynamically with the test environment. If they behave as expected, then the design specifications are confirmed: otherwise, investigation should follow, and changes may be needed to the subsystems design specifications. The decisive test, as always, concerns itself with the ability of the SoS to neutralize all of the symptoms of the original problem. If a significant time has elapsed, allowing the original problem to morph/evolve, then the decisive test may be that of solving the evolved problem: SM practitioners will have been monitoring the changing problem situation and

Systems Engineering: A 21st Century Systems Methodology Derek K. Hitchins
 ISBN: 978-0-470-0585-5

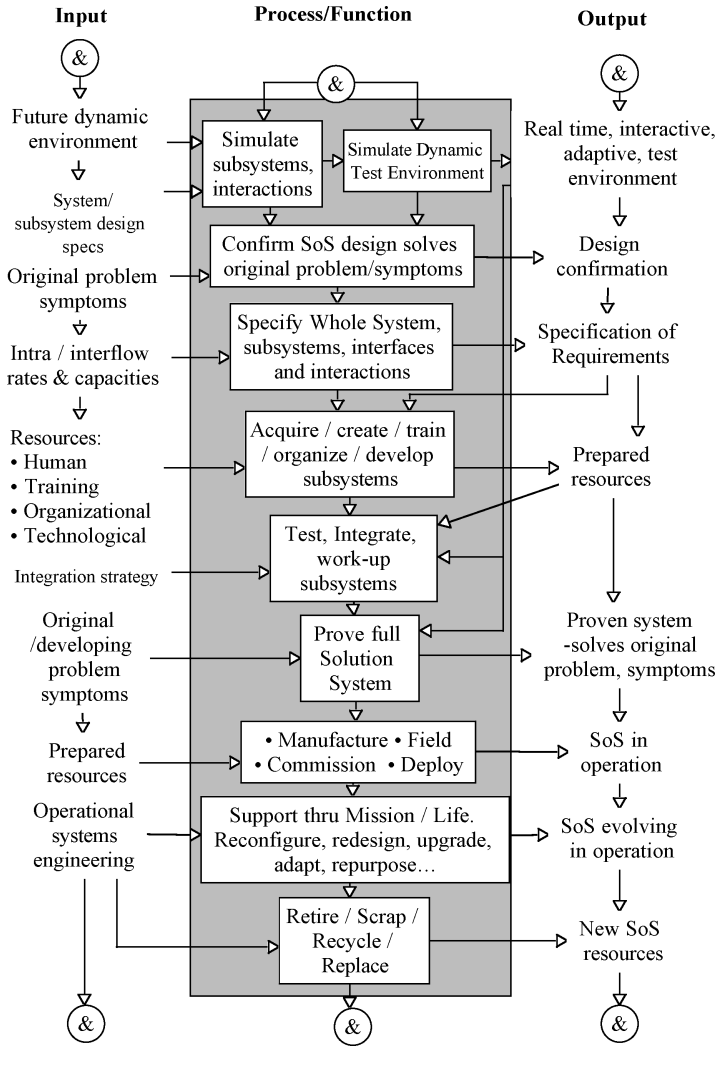

Figure 12.1 Create, prove and set solution system (SoS) to fulfill its purpose.

may/should have allowed for such changes in the developing designs. For instance, the SoS design may incorporate the ability to redesign itself . . . .

Requirement Specifications

Once confirmed, with any changes incorporated, the design specifications may be transformed into a complete 'matched set' of requirement specifications, which should specify, in detail,

and with appropriate precision, all aspects of the whole solution system, of its subsystems, of the interaction infrastructure that binds and couples them, of any interfaces within the solution system, or between the solution system and the external world. And, since we are specifying the requirement for an open, adaptive, interactive solution system, it is important also to state the solution system's purpose, the problem it is seeking to solve, the CONOPS, containing system, interacting systems, contexts and scenarios; else, the designs and the specifications will be incoherent.

Manifesting Different 'Kinds' of System

The next stage in the procedure is to realize the various subsystem specifications. How this is achieved will depend on the nature of the subsystem. Some subsystems may be comprised of people operating as a team, a Human Activity System (HAS), such as a management team. Others may be people operating as a team with technological support, such as an Air Traffic Management (ATM) aircraft flight control team, where controllers operate largely via their radar consoles, control panels and radios, to identify, track, direct and ensure the safe passage of transport aircraft: this would be considered a sociotechnical subsystem. Some solution systems emerge as processes, with groups of subprocesses/activity groups, to be organized and implemented. Yet again, there might be subsystems that are purely technological, such as an automated chemical process plant, an intelligent robotic device for cleaning radioactive contamination, or a radar receiver, part of a complete radar.

The manner of realization is dependent on the nature of the system or subsystem. For HASs, it may be appropriate to recruit people with appropriate skills and experience, bring them together in the working environment and train them to work together as a team: this does not always work; sometimes an individual is not a team player, and will not fit in. Being part of any team involves communication, coordination and cooperation with other team players to promote synergy: team players are open, interactive and adaptive subsystems. . . as a result, the 'performance' of a team can develop and improve over time, and the team can adapt, develop and improve as it undertakes different tasks and interacts with other, different HASs.

For HASs, subsystem synthesis involves configuring, organizing, managing, educating, training, encouraging and leading individuals to manifest functional capabilities as required by the design requirement specification. For a maintenance organization, this might mean establishing teams of trained mechanics, technicians and engineers to detect and locate defects and failures in some operational system, to replace defective parts, to confirm the repair, to 'mend' the defective or faulty part, and perhaps to service the operational system according to a prepared schedule. There is, then, the concept of a process, or procedure, that the people follow so that the team as whole performs the necessary functions and exhibits the requisite emergent properties, capabilities and behaviors. (This is analogous to the implementation of, say, an electronic subsystem design where signals/data flow through a series of transfer functions, organized in such a way as to perform some function, part of the whole's emergent properties, capabilities and behaviors.)

For a military command and control system, synthesis would involve the formation of teams of, hopefully experienced people to undertake intelligence gathering, intelligence analysis, operations analysis, tactical and strategic planning, logistics management, decision support, decision-making, execution, liaison, etc., etc. Within each of the teams, a process would be developed and evolved for undertaking the various tasks, and for achieving the objectives of the team — which contribute, of course, to the goal of the whole command and control system. Synthesis would include developing

the processes, and practicing their employment to work-up team capability; it would also include coordinating and communicating between the various teams, which would also be practiced and worked up.

The same approach to subsystem synthesis can be true for sociotechnical systems, such as the ATM flight control team. The behavior of each team member may be circumscribed by interactions with the technical elements — the integrated control console, with radar and data displays, controls, communication systems, identification systems, track labels, warning systems, procedural software, etc. So, although the human members of the team are potentially flexible and adaptable, the degree of such flexibility is intentionally limited by the technology: lives may depend on coherent, predictable and repeatable behavior on the part of the controllers.

So, the realization of a 'new' ATM flight control system might not rely entirely on the human ability of the team members to develop their team performance over time; on the contrary, team performance is required to be at top level from the word go, and the technology is employed, partly at least, to ensure that required consistency of performance. The procedures and processes built into the ATM 'machinery' must be, and are, thoroughly tested and tried. In practice, this results in controllers becoming very familiar with particular ways of working, and reluctant to embrace (risk?) change. In effect, their potential for adaptability and flexibility may be reduced. If and when some new technology or process is introduced, it may prove necessary to go through the process of working-up the team, i.e., enabling them to try out the new facilities, to become familiar and proficient in its use off-line, as it were, before 'going live.' In so doing, the experienced team members will adapt to the new system/technology, but may also find faults or defects in operation, process and procedure.

For other sociotechnical systems, the situation may seem different. A fighter interceptor aircraft with a crew of one, or two, say, forms a seemingly small sociotechnical system, until it is realized that, while airborne at least, the crew is in close communication with other fighters, ground controllers, airborne warning and control aircraft, naval ships, ground command and control, military forward controllers, etc., etc: so, very much (part of) a highly dynamic sociotechnical system. . . . When designing, developing and constructing the aircraft, on the other hand, it may seem to engineers in particular that the human connections are rather localized; after all, most of the internal subsystems are technological: engines, powered flying controls, generators and alternators, power distribution, fire-and-forget air-to-air weapons, counter-measures, and many, many more.

An alternate view of the fighter interceptor aircraft is that it performs *only* as a sociotechnical system. Without the crew, it is an inanimate object; with the purposeful crew, it becomes a synergizing part of a team of such aircraft, and it becomes dynamic, open, interactive, adaptable, and even innovative. So, another way of looking at the technology is that it extends the capabilities of the crew beyond their human limits. With the aircraft, the crew can travel faster, climb higher, see further, 'throw' further, see threats earlier, and so on. Looked at in this way, the technology can be seen as making the whole aircraft better able to contribute to the social group of air defenders; so confirming the view that the fighter interceptor is a sociotechnical subsystem, although one, perhaps, with many technological sub-subsystems. . . .

Since the systems methodology concerns itself with realizing purposeful systems to solve problematic situations, solution systems will seldom be other than social or sociotechnical systems: few machines exhibit purpose, conceive of a mission, and pursue that mission purposefully of their own volition; in the future, intelligent robots will, no doubt, do so; but not today. The machines that we build at present tend to be artifacts; tools made by humans to be used by humans. It is the humans who have the purpose, while the artifacts are at best purposive, i.e., an observer may ascribe purpose to them. A hammer, for example, has no purpose. An observer may suppose it is

for hitting things — he or she ascribes purpose to it. A workman may use the hammer to achieve his purpose of driving a nail. However, he may also use the hammer to prise open the lid of a paint-pot, or to stop a piece of paper blowing away. Together the man and the hammer form a small sociotechnical system, in which the adaptability of the man extends the use of the hammer. (Hence, perhaps, the old saying: 'If the only tool you have is a hammer, soon everything starts to look like a nail.')

Organizational considerations

The various subsystems comprising the solution system may be created at the same time in the same location, perhaps as a single project; where the solution system is largely managerial or technological, that would not be unusual. On the other hand, the whole may be developed as a number of parallel projects, with each project, perhaps, creating one of the major subsystems. This may be more appropriate where the whole solution system is large and diverse; where the subsystems are significantly different in nature from each other; where the technology of each major subsystem is significantly different; and so on.

The projects may be independent, although that carries with it the risk of so-called specification 'creep,' such that the dynamic emergent properties, capabilities and behaviors of the various subsystems no longer constitute a matched set; this may not become evident until they are brought together for final test and integration. To guard against that, it may be prudent to retain a coordinating function, which regularly audits the emerging subsystems to see if they are conforming to specification. Sometimes deviation from specification can be rectified; sometimes it is unavoidable. In the latter case, there may be a need to rejig the whole design; tests using the dynamic test environment may guide the decision. Where people are part of the various subsystems, they may adapt without further ado, or may need further instruction or retraining.

Integration considerations

When the subsystems have been formed, by whatever means, they may be brought together, or integrated, with the dynamic test environment acting as the watchdog on the success of proceedings. Even at this late stage, and in spite of all previous efforts, it is not unknown for errors to emerge; some may be ignored, if minor; some may be repaired 'on the fly' and incorporated into the specifications; others may be 'showstoppers,' and may require significant redesign, if not abandonment of part or the entire project. This can arise in technological solution systems, for example, where all functions seem to be operating correctly, but the timing and coordination of interactions between functions is incorrect, erratic, etc., and cannot be corrected: it is to discover this kind of error, in part, that dynamic testing is important particularly in the final stages.

There are those who do not favor such final integration and test prior to deployment. One alternative approach is to transport the various developed parts of the whole to their operational site, and to integrate them there, going straight to operation. It can be tempting, since it seems on the face of it to save time and money. It may also prove risky, as witness the Hubble space telescope, where pre-launch test of the optical system would have revealed defects that in the event were not discovered until switch-on in orbit: such approaches may turn out to be false economies.

Instead of fully testing the whole SoS prior to setting it to work, it is also possible to 'work-up' a solution system in situ, and in 'live operation.' Here, all the various subsystems are delivered to their various sites and interconnected so that the whole operates. It may not work optimally, however, so there may follow a series of 'adjustments' to individual subsystems and to their intercommunications, to enhance whole system operational performance. Much of this is unlikely to involve technology in the first instance, since that tends to be relatively inflexible; instead, the people-elements adapt their behavior, refine their processes and procedures, adjust, or even change tactics, etc.

For complex sociotechnical systems, then, there may be a two-stage process for integrating the various subsystems:

- In the first stage, technological facilities are integrated progressively, joining perhaps two parts initially, and ensuring they work together correctly, before adding a third, and then a fourth, and so on. This progressive integration helps to avoid the situation where all the technological parts may be brought together and connected in one go: thereafter, isolating the source of faults and defects may prove difficult-to-impossible; hence, the progressive approach is designed to aid fault detection and isolation. This integration and test is best undertaken where there are facilities for effecting any redesign, repairs and reprogramming. . . .
- In the second stage, all of the subsystems, social, sociotechnical and technological, are deployed to their operational situation, comfortable in the knowledge that the technological parts at least should work as required. The human elements — the people systems — may now be encouraged to interact with each other and with other systems in the operational environment. The whole solution system will then take some time to adapt to the situation, as the human elements accommodate the changing situation, and as the whole, open, interactive SoS adapts — as it was designed to do.
- This is, in effect, post-delivery optimization — some would call it operational systems engineering — see Figure 12.1, and is common practice in, for example, multilateral forces, where force elements are brought together from different nations and expected to work as a unified whole. It also happens, on a smaller scale, when design teams are formed at the start of some major new project; team members may be drawn from many different sources, and may have many different backgrounds, experiences, methods, techniques and procedures. See also Case C: The Total Weapon System Concept.

A third example might be that of the communications infrastructure for a major military intervention such as Kosovo, where there was a complex mix of military and nongovernmental organizations (NGOs), including the Red Cross and the Red Crescent. Many different organizations were continually arriving and leaving, such that there was — and could be — no established communications infrastructure. Instead, the structure was in a permanent state of flux: this *was* the optimum state, as the infrastructure adapted its links, protocols, switches, directories, etc., to the turbulent need.

The proven solution system may be a one-off, or reproduced many times. Typically, a command and control system, a new political party, or a new enterprise might be a one-off, while a new car, tank, ship, aircraft or avionics system might be destined to be replicated many times; the latter to be installed in all of the aircraft in a fleet, or on a production line. Conceivably, then, there may be different sociotechnical solution systems, each with the same technological content, each pursuing different purposes and performing different missions: the human element in each sociotechnical system employs the same technology for different purposes.

Operational systems engineering — continual optimization in operation

Once proven, the SoS may be put into operation, interacting with other systems in some dynamic operational environment. The operational SoS, as we may now call it, will not remain constant, however: the problem it has been designed to solve may be continually morphing, and the SoS will have to adapt and evolve to match; the environment may be changing, and new technology may become available which, if introduced, might make operational behavior, support, or resourcing better, less expensive, less polluting, more secure, less energy dependent, etc.

The SoS may 'enter service,' i.e., become operational, without the design having taken account of later changes to the original problem; this is not uncommon practice where the SoS has major technological components that have may have taken some time (a decade or more would not be unusual) to develop and construct. In the short term, the human element of the delivered SoS may be able to accommodate the difference between the SoS as it is, and the SoS as it should be to address the morphed problem. In any event, there will probably be need to 'improve' the SoS, to catch up with the changing situation/problem. Unless, that is, there are insufficient resources in the kitty for the improvement; or, perhaps, the owner of the new SoS decides to continue with the SoS as it is, considering it 'good enough.' In essence, the owner may choose to resolve the problem, rather than solve it as originally intended with the concomitant expense, and the potential loss of capability during upgrade. . . .

If the SoS is able to continually reinvent itself, through redesign, reconfiguration, etc., then the SoS may survive indefinitely — although not necessarily in the same form as it started. If the SoS can adapt and evolve as fast, or faster, than the environment in which it exists, then it has the prospects of lasting indefinitely. If not, like the dinosaurs, it may die out. . . or be supplanted by a better-adapted species.

Component of the Whole SoS

The whole solution system consists essentially of three elements — see Figure 12.2. In practice, these are unlikely to appear as three discrete subsystems. Instead, they will be 'woven into the fabric' of the overall solution system:

- The mission/operating system, that which interacts directly with other systems in the operational environment.
- The viability management system, which ensures that the mission system is capable of pursuing the mission. Conceptually, the viability management system provides a platform, or base from which the mission/operating system may operate securely and safely.
- The resource management system that ensures the availability of all resources, to maintain the mission/operational system, the viability management system and itself, the resource management system

The notion of the three complementary elements of the whole SoS is universal. Apollo would not have reached the Moon and back without the ground facilities around the Earth for communication, telemetry, recovery, etc. The Apollo crews would not have been proficient without crew training facilities, both for working together within the craft, and for undertaking extravehicular activities in space to inspect the exterior, make repairs, etc. The complete mission system would not have been

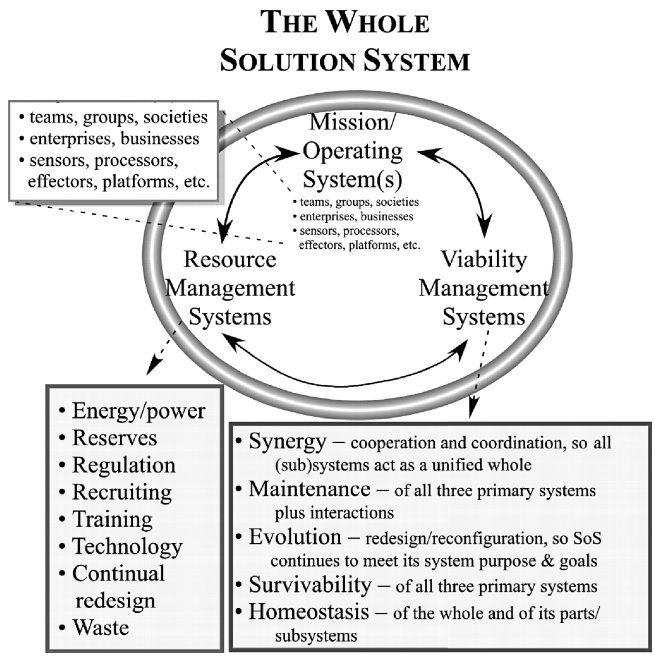

Figure 12.2 The major components of the whole SoS. In addition to the Mission/Operating system, i.e., that which directly addresses the task of solving the problem, the systems methodology generates the Resource Management System, to ensure that the Mission Operating System is continually/continuously resourced, and the Viability Management System, which both maintains the whole system capabilities and evolves/adapts them in line with developing needs and threats. All three components combine to assure that the Mission Operating System continues to pursue its mission and purpose, conceptually endowing the SoS with unlimited life.

viable without maintenance, servicing, telemetry and facilities to test for defects and failures. The public may have seen the rocket and the space vehicles — the mission/operational system — but the other elements of the 'triad' became highly visible when things did not go according to plan.

Considering an individual person as an archetypal operational SoS model:

- the brain, senses and motor controls might be viewed as the Operating System, collecting information, strategizing, planning and executing 'missions';
- the body with its skeleton, organs, nerves, immune system, etc., might be seen as the Viability Management System, providing a dependable base from which to mount operations;
- and the Resource Management System might be comprised of:
  - the food and liquid intake, processing and waste disposal elements,
  - blood circulation, carrying oxygenated blood to the tissues and muscles (energy distribution),
  - returning venous blood to the heart and lungs (waste management), and so on,
  - all needed to support the Operating System and the Viability Management System — not forgetting the Resource Management System itself.

In creating new systems, it is not uncommon to concentrate on the mission/operational system, to the detriment, perhaps, of the other two. The whole SoS may be created under a number of different project headings such that the mission/operating system is started first, followed later by the viability management system as a series of disjointed projects, and later still by the resource management system. It can make better sense, however, to develop and introduce all three as a unified whole.

For example, when introducing a new fleet of passenger aircraft, it is important to introduce at the same time all the support facilities for maintenance and servicing, for training the maintainers, for training the crews (including simulators, part-task trainers, etc.) and the requisite logistics support system to ensure availability of consumables, facilities, spares, etc. Where a new SoS is being added to an existing catalog of systems, it may appear to make economic sense to rationalize the components of the SoS such that they are compatible with those already in use. On the other hand, it may prove much quicker and less costly to keep the whole SoS support and resourcing independent of other systems.

Summary

The processes of creating the solution system and its parts are conceptually straightforward, provided the SoS design has been successfully and comprehensively achieved. To confirm this, it is prudent to confirm the design by simulating the requirement specifications for the various subsystems and their interactions, and by setting the simulated SoS to interact with an open adaptive, representative interactive environment.

Once confirmed, subsequent creation involves synthesizing the various subsystems and their mutual interaction infrastructure, and then bringing the realized parts together and testing them dynamically and interactively. The acid, or decisive, test as always will be the ability of the realized, dynamic, interactive, adaptive SoS to eliminate all of the symptoms associated with the original problem. Where the problem has morphed since the original analysis on which the design is based, there may be a need to update the design and SoS before putting it to work; else, concessions may be made and any capability shortfall made up later.

The way in which subsystems are synthesized will vary according to the nature of the particular subsystem. Solution systems and their subsystems are generally either human activity systems (HASs), such as teams, groups, societies, communities, enterprises, etc., or sociotechnical systems where people are supported/enabled by more or less technology, with continuous/continual interaction between the two. Within the technological elements, there may be discrete sub-subsystems that are entirely technological; the technological sub-subsystem will be open, interactive and may even be adaptive. . . .

The whole SoS is realized by bringing together the various synthesized subsystems, causing them to interact, and setting the whole to work in its operational context, as an open adaptive, interactive whole. There are arguments in favor of performing the integration of the various subsystems in a nonoperational context, using a simulated operational context; this has the advantage that any errors, defects or faults that emerge may be addressed more readily and with less risk in this nonoperational environment. It may also be possible to work up team performance offline, in this manner, to avoid the problems of deploying an immature solution system.

The systems methodology conceives, designs and creates whole solution systems, including not only the operation system, but also the viability and resource management systems. If these are manifested as the whole SoS, then the SoS will have the ability to continually redesign itself, to

adapt to changes in the problem, or even to quite different needs. Such a system may be intended for just one mission, or for a lifetime — where, if the SoS is sufficiently adaptable, the duration may be indefinite.

Assignments

1. You are concerned with the urgent delivery of a complex, and very expensive, technological system to be launched into near Earth orbit, where it will be used for remote sensing of Earth resources and for observing the Sun. All the various subsystem elements have been developed and tested separately: all meet their specifications. Time does not permit full test and integration of all the parts prior to meeting the next launch window, in one month's time, and the next feasible launch window is a year later. You have a choice, then, between proceeding with full test and integration and missing the window, or installing the parts directly into the launch vehicle, to meet the launch window. Or, perhaps there is a third way? List the factors you would take into consideration in making your decision, and justify your choice, as though you were presenting to a congressional appropriations committee — who will wish to hear neither detail, nor 'techno-babble!'
2. Conceptually design a self-healing human activity system, i.e., one that is capable of repairing itself. Explain and justify your design. Conceptually design a self-healing technological design for a system of your choice. Explain and justify your design. In both instances, explain how you would synthesize and prove your designs, and where they might be applied in the real world.