11

SM6: Optimize Solution System Design

The optimist proclaims that we live in the best of all possible worlds; and the pessimist fears this is true.

James Branch Cabell, 1879—1958

Approach

Optimizing a solution system design means making it as good as it can be in context, particularly in terms of its performance and effectiveness. Optimization, then, enhances and refines, so that optimization may be seen as part of, or perhaps an extension to, solution systems design, making the design even more relevant to the situation in which the SoS will find itself.

As we have already seen, from 'The GRM and the Systems Approach' on page 135, optimizing can seem a complicated business. The system-to-be-optimized interacts with, and adapts to, other systems in their mutual environment, and vice versa. So, change the effect that system A has on System B, and System B is indeed changed. But, that means that the effect that B was having on A may also change, perhaps reinforcing the initial change, perhaps neutralizing it, or perhaps doing something quite unexpected. (This is Newton's Third Law at work, or more generally Le Chatelier's Principle.) And that is just with only two systems interacting; suppose there are three, or four, or a dozen other systems, all mutually interacting dynamically, all changing- and being changed by the others. (For the implications of Le Chatelier's Principle see Understanding open system behavior on page 12, Organismic control concepts on page 20, and the Principle of System Reactions on page 54.)

It would be easy to throw one's hands in the air and declare the problem too complicated to address. Some systems engineers actually disparage optimization, declaring that it leads to 'gold-plating,' i.e., making the solution system expensive and overblown: interestingly, that should be the precise opposite of the truth. Functional, or performance optimization is one of the issues central to the practice of systems engineering; walking away from it is not be a sensible option.

As a less-than-ideal alternative, systems engineers may conduct some tradeoff analysis, considering the solution system as a closed, isolated system. The idea is to generate a range of solution

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

options, and a set of criteria by which to compare the options, and then to conduct a tradeoff to see which of the options best fits the criteria. The problem lies in knowing whether or not the statically selected option will actually deliver the goods when the SoS interacts dynamically with other systems; chances are, it will not. But, suppose it were possible to adjust the system-to-be-optimized while it was operating interactively, and to conduct tradeoffs at the same time, so that the changes in dynamic behavior, performance and effectiveness could be observed, and the 'best' results applied and retained . . . .

Methods, Tools and Techniques

Cost and capability

By this stage of the systems methodology, the 'nature' of many of the design subsystems will be evident. For example, it will be evident that many of the prime mission functions can be conducted by people, or perhaps by some technological system or artifact. While the nature of the subsystem might be evident, the specifics of each subsystem will not be evident: it might be clear that some kind of missile will be needed, but not type or range; radar will be needed, but not which kind, transmitter power, receiver sensitivity, etc; a team of sappers will be needed, but without knowing how many in the team, and their capability; and so on. Part of the optimization process concerns itself with filling in these blanks: at the end of the process, we should know the 'emergent' specifics of the various subsystems.

This presents a problem, in that the process has to be able to 'try out' — in simulation — the effects on the whole of having a bigger, or smaller, missile; a more-or-less powerful transmitter; a more-or-less capable team of sappers, and so on. It may be important, too, to know the relative costs of having more of this, or less of that, since the measures of effectiveness (MOEs) that will be used to evaluate the SoS may well include cost-effectiveness, cost–exchange ratio, profitability, return on capital employed (ROCE), and so on.

One way of addressing this issue is to derive models of such subsystems that relate performance/capability to likely cost. Analysts, who can forecast how much a future aircraft is going to cost by extrapolation from past and present aircraft costs, develop such models routinely. Similarly, there are predictions for growth in computing power, range of weapons, cost of avionics, etc., etc. There is, indeed, an industry dedicated to such speculative parametric analysis, and we can use their output in the optimization process to predict both the capability and cost of, particularly, technological subsystems.

Optimizing the whole

It can be shown that optimization is practicable only when addressing the whole system; trying to optimize part of a system may well result in deoptimizing the whole. Without going into the theory, this is apparent from the orchestra analogy: optimizing, say the brass section of the whole so that it gave the best, and possibly therefore the crispest and loudest, rendition would unbalance it, compared with other sections, and would deoptimize the whole orchestra. It is, indeed, an interesting observation that an optimum solution (orchestra) is generally comprised of suboptimal parts (sections); such an observation is unlikely to impress the musicians! Unless, that is, they realize, that to be suboptimal, in that context, is to listen and adjust your playing so that it is compatible and harmonious with that of other musicians in your own and other sections. In musical terms, an optimal musician might be a soloist, which may render her or him unsuited to play in an orchestra.

Recalling the analogy of the Formula One car made up from the best bits of other cars, it is unlikely that putting the best parts from different cars together will produce an optimum/best result, if only because the parts have not been designed to work together in a balanced, harmonious way. Gear ratios that worked fine with one car might prove less than ideal for the composite car. Traction controls that permitted maximum acceleration at the start of the race in another car, might not give best results in the composite car; and so on. Note, in both analogies, that optimum performance is observed only under testing, dynamic, operational conditions — just where you need those emergent properties, capabilities and behaviors, and just when the whole has to be greater than the sum of the parts.

Seeking an optimum configuration can seem difficult. Consider a system comprised of a number of interacting subsystems: changing the properties of any subsystem is likely to impact on other subsystems, and the change induced in them will both feed back to the initial subsystem and may feed forward, with variations, to other subsystems. Essentially, making a sudden change can result in a series of reverberations, and as the parts reverberate, the emergent properties, capabilities and behaviors of the whole system are changing, too.

Too far fetched? Things don't work like that? Consider the effect of a sudden hike in the price of petrol/gas on a city's commuter transport system. First, there will be protests and outcries, while at the same time long queues will form at the gas stations as people fill up and some attempt to hoard fuel. Next, there will be a marked increase in people commuting into the city by train, and a significant reduction in traffic on the roads. Increased fuel prices will be passed on to retail commodity prices, so people will use their cars to go to the mall, or the supermarket, and stock up on foodstuffs, causing local traffic chaos at the malls, food shortages, panic, and an increase in local traffic accidents.

Gradually, as the raised price persists, and noticing the reduction in commuter road traffic, some people will go back to using their cars for commuting, flirting perhaps for a short period with car-sharing, but eventually going back to solo car travel — after all, they prefer the freedom of being isolated in their own car in a 10-mile traffic jam to sitting or standing on a commuter train crowded with other people. Meanwhile, there is a slow drop in the price of fuel as politicians realize that reelection is looking less and less likely. Next there may be an over swing, as people who usually commute by train decide they have had enough of the exacerbated overcrowding on trains, so they switch to the roads, causing enormous traffic jams. And, finally, things settle back to where they were originally. . . , but with a slightly raised cost of fuel.

Eventually, the commuter transportation system will return to an uneasy dynamic stability, with some overcrowding on the railways and some traffic jams on the roads and highways. It is hardly an optimum, but it is a kind of balance — perhaps 'the best in the circumstances.' And the system illustrates the optimization issue. If you change any one subsystem in the interacting set of subsystems, it will affect the other subsystems, and they in turn will affect it and each other.

One solution to sorting out the tangle is dynamic simulation, not only of the system-to-be-optimized, but also of the other systems with which it interacts at the same time (which is the systems approach, of course). Given such a dynamic systems simulation, it is possible to change any one of the subsystems, to observe its effect on the whole, the interactions with other systems that change everything, and eventually to observe the new point of open system stability — all in simulation, of course; it would be generally be far too expensive, disruptive and time consuming to do anything else. But, how do you find the *optimum* configuration?

One way is to design the solution system very much along the lines of other, existing systems. This has the advantage that it tends to reduce risk, and it employs familiar skills and capabilities: on the other hand, it inhibits innovation.

A second approach is to alter a subsystem or component (easier in simulation), observe the effect on the whole, in terms of, say, cost-effectiveness, and to set the component parameter value to that which gives the greatest increase in effectiveness. Having done that for one subsystem or component, the process may then be repeated for others, gradually improving the chosen measures of effectiveness (MOEs) each time. Although this is a pragmatic approach and can give reasonable results, a little thought may suggest that it need not give the best result, since each change alters the point of balance between the interacting subsystems, so that each change can affect previous changes, rendering them no longer optimal. The situation can be ameliorated by 'going round again,' i.e., by repeating the process of altering each subsystem or component in turn to find the overall best result in context — the optimum.

Another approach is to employ cumulative selection to cut the Gordian knot of complexity. Suppose we have a system, Blue, with N subsystems, operating and interacting with other systems, and the objective is to find the optimum configuration, where optimum is identified as, say, achieving the greatest effect on one or more other systems, Red, in the simulation. We can start by changing each of Blue's subsystem in turn by a small amount, measure the change in effectiveness, and then return each subsystem to its original setting. At the end of N changes (for N subsystems), we should have a record of the changes in effectiveness cause by each subsystem change. One of the changes will be greater (inducing greater effectiveness/performance improvements) than all the others; incorporate that selected change to that subsystem, leaving the others at their original setting. Then repeat the exercise, each time increasing the overall dynamic effectiveness of Blue by a small amount, until it cannot be increased any more.

The result of this cumulative selection process should be the identification of the best configuration of Blue to give it maximum effectiveness while it is operating and interacting with, and adapting to, other systems in their mutually open system environment. There can be problems with such a simple method, however. . . it is possible to find 'local maxima,' i.e., local optima that trap the optimization process and prevent it finding the overall maximum. Furthermore, although the cumulative selection process might be undertaken manually for a very few subsystems, it would become very cumbersome for a large number of subsystem, interconnections, etc.

It is possible to liven up the process by using so-called genetic methods. Using this approach, the design is replicated using 'genes' to 'code for' different aspects of the design; there might be a gene to code for a function, or more likely to code for the 'magnitude' of a function. Another gene might code for the existence, or not, of an interaction. Yet another might code for the capability of some subsystem or part, so that the numerical value of the gene corresponded to the degree of capability (speed, power, energy, reliability, efficiency, etc., etc).

Given a comprehensive set of genes, it is then possible to create an instance of the solution system in a dynamic simulation, to test this solution system instance in operation, and to evaluate its effectiveness when interacting with other systems, also represented in the simulation. The 'genome' can then be 'evolved' to find the optimum design: examples follow.

Disaster relief example

Suppose that the system of interest were an international disaster relief organization, and the objective was to set up a disaster relief project with the purpose of saving as many lives as possible within the disaster area, within a given amount of time. The simulation might then represent the disaster area, Red, with its geography, damaged infrastructure, floods, landslips, dead and dying of thirst, starvation and disease, etc., and it would also represent a notional disaster relief organization with genes coding for agency staff, vehicles, bridge builders, helicopters, field hospitals, secure storage depots, escorting troops, etc., etc; in effect, the full set of genes would code for all of the elements of Blue, the disaster relief organization,.

On first running the simulation, the disaster relief operation would produce initial results, with some lives saved, together with some resources wasted, and others not available in time or location. (A useful way of simulating the in-country activity might be to use intelligent cellular automata, moving, acting and interacting across a terrain map, set to present the contemporary, disastrous, in-country situation.)

The (simulated) performance of the disaster relief organization could then be improved and optimized by providing more bridge-building facilities, perhaps, or by providing less in the way of, say, bedding and shelters, but using the transport to convey more food, potable water and first aid instead. To find the optimum mix, the following procedure might be employed.

Randomize all the ‘gene’ values, run the simulation, record the results (numbers of lives saved within the set period). Randomize all the genes again, and repeat the exercise. And again; and again. Repeat, perhaps, one hundred times or five hundred times, recording the results each time. Then compare the results and see which combination of genes resulted in most lives saved in the least time. Incorporate this particular gene combination as the new baseline systems design set: repeat the exercise for another one hundred runs, varying the genes randomly about their new baseline values; and so on.

Gradually, and progressively, the simulation accumulates the combination of genes that saves the most lives in the least time. Stepping through the simulation, observing how it works, and understanding why it works better than other configurations can confirm whether, or not, this result is optimum. (Sometimes the results of such simulation runs can be counterintuitive, but are likely to be broadly correct, nonetheless — always provided the simulation is accurate, current and complete.) If the original problem facing the systems methodology was ‘how best to organize and equip the disaster relief force,’ then the simulation should provide the acid test of the solution system design — does it eradicate the symptoms of the original problem? So, this genetic cumulative selection process would simultaneously justify and optimize/validate the overall solution system design.

The process can be widely applied, and is by no means limited to disaster relief: it is also capable of development. Once an optimum design has been proved, new genes can be introduced to code for facilities that might be considered ‘potentially useful;’ to see if their inclusion would be justified. Some genes may be driven towards zero values, indicating that they are adding nothing to the design performance of the overall solution system. Affordability can be included, so that optimization might focus on maximizing the ratio of effectiveness to cost, so giving a ‘best value for money’ solution; or, perhaps, optimization might be aimed towards minimizing casualties in war, enhancing cost–exchange ratios, etc. As many disaster relief organizations are charities, or nongovernmental organizations (NGOs), the way in which they fund projects can form part of the simulation, using publicity from ongoing disaster relief work to raise awareness and encourage donations; this ties together dynamic interactions between the disaster-stricken country and the nations trying to bring relief: and so on.

The naval destroyer example

Figure 11.1 shows the overall procedure within the systems methodology for optimizing whole solution system designs. The process is, in practice, easier to execute than to explain. We have seen how one might optimize the design of a disaster relief operation. Consider, as a different example,

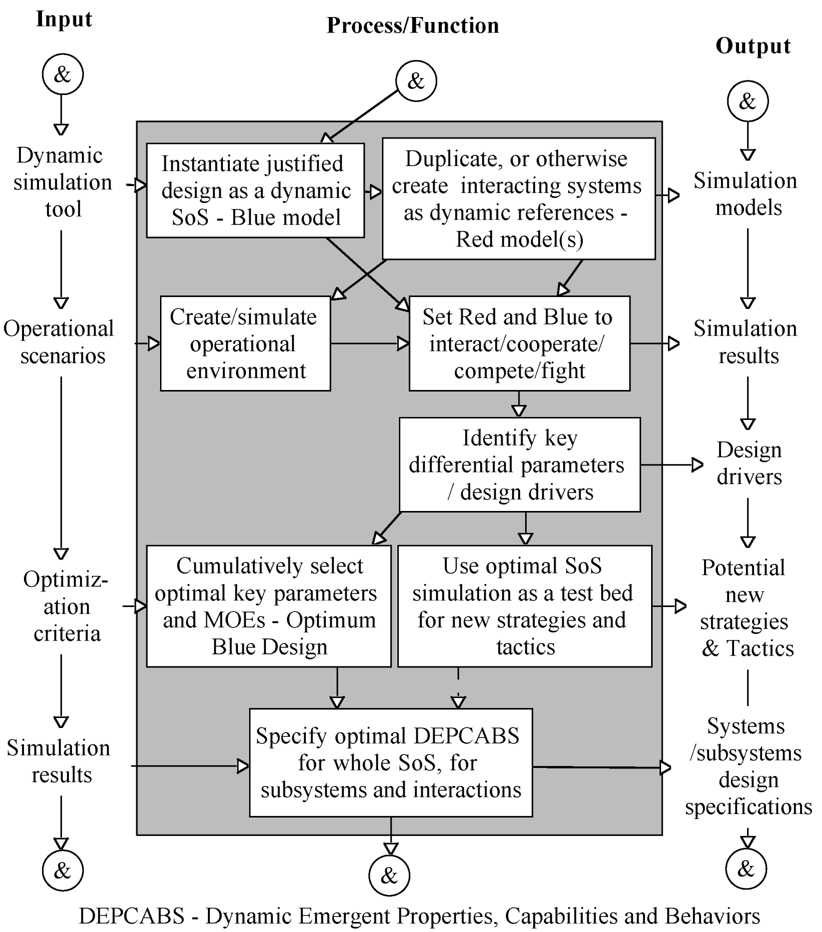

Figure 11.1 Simulation-based dynamic optimization of the solution system design. See text.

how it might be possible to optimize the design of, say, a naval destroyer. Unlike the disaster relief organization, there is no defined Red system, or set of systems, with which to interact. The new destroyer might come up against a wide variety of future opponents, and little is known about future opposition at design time; intelligence on such matters is notoriously uncertain. So, how to proceed. . . ?

One way is to develop a robust dynamic simulation of Blue (see The GRM and the Systems Approach on page 135 *et seq.*), the new destroyer, using as numerical values for the instantiation taken from current destroyers and current equipments and facilities, such that the destroyer performance would be likely to match that of contemporary vessels. Once that simulation model is tested and established as Blue, it can then be duplicated and nominated as Red, an opposing destroyer. Red and Blue can then be cross connected within the simulation such that each can potentially sense the other with radar, communicate with the other via radio and data link, intercept and exploit the other's transmissions, fire at each other when within range, inflict and experience damage, attempt to repair that damage, run out of ammunition, etc. The environment for both ships would include sea states, radio transmission characteristics, and so on. The crews for each ship would be presumed trained to the same degree, operating to the same doctrine, so that command and control for each ship would make the same decisions under the same circumstances.

Once the model had been set up and verified, Red and Blue could then be set to combat, sensing each other at sea, approaching to within engagement range, and then engaging — or not — according to the rules of engagement (ROE). Each ship may engage the other, may inflict damage, and may receive damage. Each ship may try to repair damage, to restore full capability. The instantaneous effectiveness of each ship will, therefore, be constantly changing. Chance and probability dictate that the outcome from any single simulation run is unpredictable; however, since the two destroyers are identical at this point, the results from many engagements should be identical — at least, they should average out to be identical over, say, several hundred simulated engagements.

Such a symmetrical model can be used initially to examine the effects of varying individual parameters. Keeping Red unchanged, individual parameters in Blue may be changed, to see what effect if any the change has on engagement outcome. Not only can material differences, such as reduced number of missiles carried, or increased radar transmitter power, be tested, but also such imponderables as the level of training, and changes in doctrine and ROE can be evaluated, too.

And, the results of such tests may not be as expected. For instance, one might expect increasing radar transmitter power to give Blue an advantage over Red, which remains at nominal power. In a typical simulation, that may be the reverse of the outcome. Why? If Blue's transmitter power increases, Red can detect Blue from a greater distance using passive radar sensors (inverse square law, as opposed to active radar's inverse fourth power law), enabling Red to fire at Blue before Blue is aware of the threat.

Similarly, it is possible to conduct trade studies. It may be possible, at least in simulation, to trade reduced training time, and reduced number of personnel for increased automation and smart technology on the missile system. The trade study would see changes made to Blue only, with an unchanged Red acting as a dynamic interactive reference system.

That is only the beginning, however. Using the same technique of cumulative selection as detailed above, Blue ship's design may be optimized such that it will engage Red with a greater probability of success. Optimization in this context requires an optimization parameter, often a ratio, to maximize or minimize. For two naval destroyers, the optimization parameter might be the ratio of Blue cost to effectiveness, the Blue–Red cost–exchange ratio, the Blue–Red casualty–exchange ratio, or perhaps some combination of all three. To make things more realistic, it may be prudent to expand the number of vessels on each side, introducing support vessels for refueling and rearming at sea, for instance, together with other war machines, surface, subsurface and airborne: see 'Instantiated layered GRM' on page 138. If there is one available, a traditional, trusted, many-on-many battle simulation may be used as a test bed on which to mount the developing, optimizing simulated Blue–Red designs

Having optimized Blue using Red as a dynamic reference, it is possible to turn the tables, freeze Blue's design and then optimize Red to be better than the improved Blue design. This may be referred to as ratcheting the designs, using each in turn to raise the game of the other. (The process has obvious risks that are readily anticipated, including those of violating the laws of physics and creating designs that cannot be built. . . .)

Optimizing supply and logistic systems designs

The notion of developing optimal solutions by setting systems designs into competition is not confined to combat scenarios. See Figure 11.2, which shows two lean volume supply systems competing for market share, perhaps on a global scale, as in motor vehicle manufacture, white goods, or brown goods. If the design and operation of a single lean volume supply system were represented in simulation, including the market with its facility to buy or not, to recycle, or not, then the simulation model of the complete loop could be duplicated and cross-connected, taking account of the availability of skilled labor, material resources, etc. Dynamic simulation of the two, initially identical, competing circles should then result in equal market share. Red may then be employed as an open dynamic, interactive reference, to make changes in the design of Blue, and to test for the likely outcome from those change over a number of simulation runs.

Using the cumulative selection technique allows optimization of the Blue design, still using Red as an open, dynamic interactive reference. Typical Blue optimization parameters, shown in the figure, might include minimum inventory, minimum cycle time, minimum waste and minimum pollution, or a combination of all four . . . .

The result of such a simulation would be the design of the optimum Blue supply circle, assuming that Red did not change — the result would be, in effect, an Ideal World design, and the differences between the present Blue design and the Ideal World Blue design could be used as a driver for change in Blue. However, in the real world, Red (always supposing there is a competing supply chain, which is usually the case) would not stand still while Blue continued to improve its design and operation: on the contrary, Red could also seek to improve its performance, to increase its market share.

It may be possible for Blue to anticipate Red's improvement using the ratcheting technique described above. First, optimize the design of Blue, keeping Red constant. Then freeze the improved

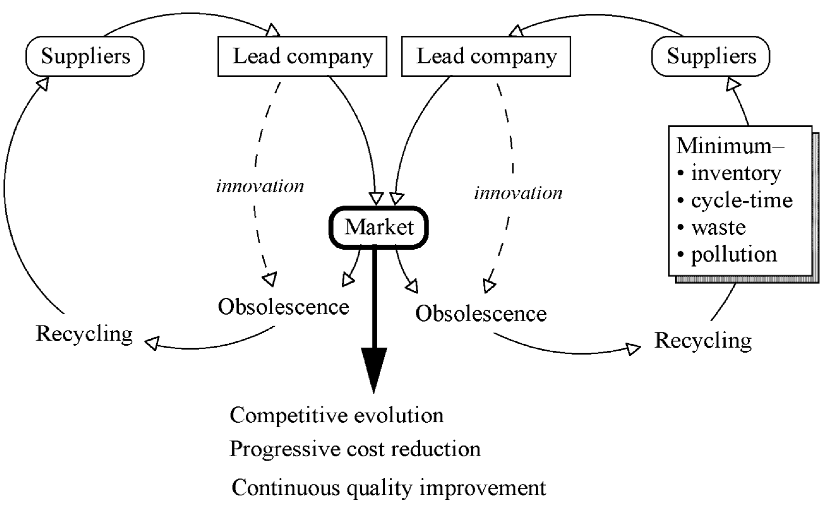

Figure 11.2 Competing supply chains/circles, Red and Blue respectively, mutually enhance each other's performance to the benefit of customers and the environment.

Blue design, and optimize Red. Finally, reoptimize Blue, this time using the enhanced design of Red as the open, interactive, dynamic reference.

In principle, this seems to anticipate changes in Red, enabling Blue to compete effectively over time. However, there may be a better way. If, instead of assuming that Blue and Red are discrete, competing, open, adaptive systems, assume instead that both Red and Blue together constitute a composite supply system, competitively providing goods into a market place or consumer system, which not only consumes, but also gives back obsolescent goods for recycling. The market will have characteristics such as the ability to become saturated, to be energized by innovative products, etc., to be more or less active according to degrees of buyers' disposable income, etc.

With this revised concept, it would be feasible and sensible to evolve the design of the composite supply system, i.e., Red and Blue as one, using the market as the open, dynamic, interactive reference. Unlike the ratcheting method, this approach has the potential to show how the co-evolution of Red and Blue might progress step-by-step, in parallel — which may be more realistic. The optimizing parameter might be different, too. Instead of emphasizing ever-leaner performance, this higher-level approach might consider, for example

$$\text{Product Utility} \div \text{Environmental Impact'} \ (= \text{'Value per Real Cost'})$$

and Energy consumption/dissipation around the make, use, recycle loop, which might have a zero sum energy goal?

Understanding the Design in Context

Once the optimization processes have been completed, the simulation models contain various parameter values which together determine the optimum solution system design, that is the best solution in the context of the environment and the other systems with which the solution system will interact in the future. So, all that has to be done now is to create a real solution system with those parameter values, and job done, right? Not right! Throughout the design process, the ability of the solution system to interact with, and adapt to, other systems has been emphasized. Those other, interacting systems adapt, too, as a concomitant aspect of the interactions. If we had tracked the value of any particular parameter, we would have noticed that it had changed during the optimization process: it may have increased, decreased, varied cyclically, or whatever.

This suggests that, were we to realize a solution system design that had been optimized while interacting with a particular set of systems in the solution space, and were that solution system to then interact either with a different set of systems, or even in the short term not to interact at all, then the optimum parameters we had so carefully identified and manifested would be inappropriate. Reconsider the example of the Blue destroyer with its active radar power being too great, so that an opposing Red ship could detect the Blue ship using passive sensors. Optimum design of Blue would not, as a result, have a reduced transmitter power, since there may be other circumstances where more power was essential (e.g., sensing through heavy precipitation or jamming, operating against ships with no passive sensors, etc.) Instead, the optimized design would enable the radar power to be varied according to situation and circumstance, with the power levels being chosen tactically by operators; perhaps automatically, such that minimum power is transmitted consistent with detecting returns; or, perhaps the degree of spread in the transmitter's frequency spectrum might be varied in concert with the receiver's correlation system; or whatever.

Similarly, optimizing the design of a disaster relief organization such that it was best suited to (e.g.) saving lives in the context of a particular disaster (hurricane on a tropical island) would not imply that the same disaster relief organization would be ideal in other circumstances (tsunami in South East Asia). Instead, what emerges from such considerations is the notion that optimization is context dependant, and since the same solution system may find itself in a variety of contexts, some of them unexpected, it follows that the solution system may have to continually re-optimize, or be re-optimized, according to situation, and at rates corresponding to the rate of change of situation.

To be Linear or Nonlinear: That is the Question

Some systems find their own optimum condition — they are self-optimizing; homeostasis also balances systems, often automatically. The various organs in the human body, for instance, all act and interact, with each 'perceiving' the sum of the others to be its environment. Notably, such organic subsystems are nonlinear, in the sense that reaction to stimulus is not always in proportion to that stimulus. This nonlinearity takes several forms. Often, subsystems for secreting enzymes, hormones, proteins, etc., have a cutoff level: no matter how great the stimulus, the output cannot exceed a certain level. Other bodily systems react, for example, logarithmically, so that they are sensitive to small stimuli, but proportionally less sensitive to larger stimuli. Closing down the iris protects the eyes so that we may see well in strong sunlight, but our eyes can also operate quite well at night with the iris opened wide trying to catch every photon. The ears similarly have a logarithmic sensitivity to sound, such that the dynamic range that the ear can accommodate is indeed wide, from the proverbial pin dropping to nearby thunder. In both instances, the sensory organs adapt to their immediate environment without apparent conscious control. Homeostasis maintains the state of a system without negative feedback: and, there is control through opposing influences or forces, such as agonistic and antagonistic muscles — again, without negative feedback, which tends to linearize systems behavior.

Overall, then, a body develops a point of balance such that all the various organs are active, within their operating range, and the whole might be considered to be in a self-organized, optimal state, with the whole alert and active in its, his or her environment. Situations arise where the sensors and other organs may be called upon to accommodate extremes in the environment, which may tend to induce extremes in some of the organic subsystems. As we have seen, these extreme demands stimulate nonlinear responses, such that no internal damage can be effected by excessive responses. Moreover, the body as a whole moves to a new point of balance, a revised optimal condition, according to each new situation, with little or no conscious involvement.

This is not to suggest that the body's nonlinear, closely couple interactive systems are perfect. Situations can arise where internal sensors can become satiated and insensitive over time, causing problems such as (late onset type 2) diabetes: optimum balance cannot always be maintained, and sufferers may collapse. And, of course, the body can be sensitive to pathogens, which may disrupt operations — hence the body's sophisticated immune system, which absorbs a significant proportion of our internal energy in maintaining our steady state.

Human activity systems are generally nonlinear, and so too, therefore, are sociotechnical systems. Consider, for example, an airliner. Its technological subsystems are engineered, and are generally linear, or at least quasi-linear. Without any crew, the airliner might be seen as an example of linear technology. Add the crew and the whole now becomes a nonlinear sociotechnical system, because its purpose and behavior as a whole are governed by the purpose and behavior of the crew. They may be using linear technology, but the way in which they use it makes the resulting operation of the whole nonlinear.

Looking at Nature's nonlinear systems, and comparing them with our manmade linear technological systems, it is evident that Nature has the edge in many areas. Nature's 'designs' have greater power density (pack more power into a smaller volume): they are often more durable, self-repairing and self-healing, too. It is possible to emulate Nature's successes by conceiving, designing and creating nonlinear technological systems, and some engineers are beginning to do that.

Interestingly, because the systems methodology follows synthetic, organismic and holistic principles throughout, the optimized designs for solution systems that it generates can be realized by manifesting nonlinear systems, perhaps with some nonlinear and some linear interacting subsystems.

Verification and Validation

The use of dynamic simulation models for evaluating and optimizing solution system designs poses the question: 'How good are the models?' It is common practice for both customers and systems engineering practitioners to decry simulation models as inaccurate, and hence invalid and not worth the money that they undoubtedly cost. To some extent, such claims to invalidity can be justified, but they also raise the question 'Is it, then, better not to model, and effectively to guess at the right answer?'

It is important to remember that dynamic simulation models, of the kind referred to throughout this book, are not mathematical models, nor are they engineering models — they are systems models, and moreover they are models of systems that are interactive, mutually adaptable, and open to their environments. If a system were to be instantiated as a team of men, a small company, or a political party, we would not expect the optimal design to be numerically precise and accurate — if it were, we would suspect that the results from the simulation had been misinterpreted. Any such team, company or party is likely to have variable numbers of people, with varying skills and capabilities which will vary over time. A team might have been 'designed' by some systems methodological process, but that would not make it a fixed entity. Teams are not static any more than the individuals comprising the team are static; instead they are developing, evolving, maturing, etc, or 'forming, storming, norming, performing and adjourning' to suggest five stages of team development (see Tuckman, 1965). Moreover, we are all aware that a good team is less about numbers, more about the right mix of personalities, skills and capabilities — as a systems capability simulation would indicate.

Simulating the design of solution systems is, moreover, predicting the future — it is inherently difficult and prone to error, inaccuracy and mistake, if only because all of the various systems of future interest may not turn out as the simulation models fondly imagined they would be.

So, while there is no excuse for inaccuracy, and certainly none for lack of rigor, the goal of validating system design models, in the conventional sense of software validation, may be untenable. It may be possible to verify models in some degree, largely by comparing their behavior with that of extant, real world systems — indeed, the models generally came from the real world in the first instance, courtesy of systems science.

There are some solution design models that cannot even be verified satisfactorily. Consider, for instance, a model that purports to show the restoration of the rule of law in a war-torn country, such as, say, Afghanistan. Suppose that courts, prisons and police stations have been largely destroyed, that there are no longer any judges and lawyers, and that police are few and far between. How might we simulate the progressive restoration of the justice system?

In some countries, it might be reasonable to simulate the dynamics by observing that the provision of sufficient money would result in builders coming forward to build the various premises, recruiters coming forward to produce the required numbers of people, and teachers and educators coming forward to teach new lawyers, judges and police — and all within reasonably predictable timescales. But, the question then becomes: would the provision of money have the same effect in Afghanistan, where the culture is different, where the law is (likely to be) Shariah law — which treats money, and the lending of money, in a way that differs from western practices, etc? Would members of the population who would make good builders, lawyers, judges and police come forward in response to money from, say, the UN and, if they would, how long might they take to do so? Apart from any other consideration, the notion that putting up money and expecting people to volunteer to spend it, is very much one of democratic, free-market economics: it may not apply in a country with many nomads, areas ruled by warlords, others by the Taliban, and so on.

Thinking about such imponderables suggests that system models for such situations must be, at best, uncertain. So, instead of constructing models with a view to validating them, mathematical and engineering style, the concept is to build simulation models as experimental learning laboratories in which, because they are systems models, the solution designs interact with, and adapt to, other systems and their environment. Moreover, such models are used to assess behavior across a wide range of situations, which may not indicate how well they might work in some situations so much as indicate those in which they may not work well, if at all — so-called areas of counterintuitive behavior (see Systems thinking and the scientific method on page 73).

For Afghanistan, the rule-of-law restoration simulation might include the facility to vary the time taken to recruit and train judges and lawyers over considerable limits, for instance, so that different runs of the simulation could assess the different possible outcomes. Running such a systems simulation might indicate the value of importing judges, lawyer and police temporarily from other Moslem nations, for instance — at least as a working hypothesis. Similarly, shortage of prisons, law courts and police stations might be alleviated — at a cost — by introducing temporary (mobile?) accommodation, the value of which could be assessed by running simulations with various options. In this way, the 'design' of the best solution is not just being tested: it is being progressively developed and optimized, too, in simulation, see Figure 11.1.

Summary

Optimizing the design of a solution system is an essential part of 'solving the problem;' optimizing identifies the right parts, brings them together in the right way, to interact in the right degree, to create the requisite emergent properties, such that the whole solution system is greater than the sum of its parts. Optimizing is, then, essential systems engineering.

Optimization can be effected only at the level of the whole system. Seeking to optimize only part of the whole may de-optimize the whole, and is likely to be counterproductive. Difficulties in optimizing the whole may be anticipated, since changing any of the parts not only affects the whole directly, it also affects the other parts with which it interacts, and they in turn may affect other parts (transitivity). Any of these direct and indirect effects may also change the whole.

Because of this potentially complex behavior, optimization often resorts to dynamic simulation of the solution system in context, interacting with other systems so that they are all open and mutually adaptive — as in the real world, of course. Change, action, reaction and transition towards optimal balance are more easily observed in simulation than in the real world, where time, cost, and risk may inhibit, or even prohibit, trial and experiment.

Even in simulation, finding the optimum set of conditions and parameters for a complex solution system design when interacting dynamically with other systems may not be trivial. The potential number of combinations may prove enormous. It may be tempting to reduce risk by designing a new system very much in the image of its predecessors, or other extant, successful systems. An alternative, innovative approach is to use cumulative selection to explore this vast landscape of possibilities, in a greatly simplified version of Nature's genetic approach: in effect, to 'evolve' a solution system design. Examples of how this can be sensibly achieved are presented in outline. (Case studies will be presented later showing this approach in some detail.)

The result of such system optimization approaches is an 'optimum design set' of dynamic emergent properties, capabilities and behaviors (DEPCABs), which are appropriate, in principle, to the whole solution system operating in a particular context. In the event, the solution system may not find itself in that particular context, suggesting that the optimum solution system design should be determined across a range contexts or, alternatively, that the solution system should be designed so that it can optimize itself according to the context in which it finds itself. This is the way in which Nature's systems, such as the human body, operate; in general they rely on closely coupled, nonlinear subsystems which have the ability to find a point of balance such that the whole body is capable over a wide range of contexts. Systems as teams of people, enterprises, etc, and sociotechnical systems, are nonlinear and can effectively self-optimize . . . . Technological systems, on the other hand, tend to be linear and less able to self-optimize in this way, although engineers are getting to grips with the design of nonlinear technological systems, which promise improved power density, flexibility, adaptability and capability.

Design optimization using dynamic simulation models brings into question the validity of such models. Systems models are neither mathematical models nor engineering models, however, and the same rules do not apply. The models of systems are, like the systems themselves, open and mutually adaptive to their environments and to other systems. As a result, the optimized system design has to be interpreted, and should result in a solution system that is also open and adaptable, rather than having one, fixed set of parameters. Given a real world situation like that in the simulated world, the adaptable solution system should be capable of operating with the corresponding optimum parameter set — whether adjusted automatically or by operators. But, for different real world situations, the solution system should be able to find different balance points, different optima, appropriate to those different situations. Solution system design then becomes concerned with how those various optimum configurations can be set up as needed, and what range and variety of situations the design should accommodate.

Assignment

A system is to be set up within a national airline to provide through-life maintenance and servicing support to a family of commercial medium-haul airliners, operating from more than two dozen airfields, large and not so large, spread across the continental US.

1. You are tasked with optimizing the design of this support system; your first challenge is to select three potential optimizing criteria, or parameters, such that the support system affords the best support in the context and circumstances.

- You should keep in mind that the cost of the support system is part of the operating cost of the airline, and as such potentially eats into profits.
- On the other hand, inadequate support could result in major, even phenomenal, expense, in terms of loss of reputation, customer confidence, revenue, and even possibly loss of life — with the consequent potential costs of compensation.

2. Your second challenge is to consider how to employ these three optimizing parameters. Should you use them independently of each other and compare the results: if so, how? Or, should you create a composite optimizing parameter and optimize the support system using this parameter?
3. Explain and justify your preferred approach.