HOW TO DO SYSTEMS ANALYSIS

JOHN E. GIBSON
WILLIAM T. SCHERER
School of Engineering and Applied Science
Department of Systems and Information Engineering
University of Virginia

WILLIAM F. GIBSON

Wiley-Interscience
A JOHN WILEY & SONS, INC., PUBLICATION

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada.

Library of Congress Cataloging-in-Publication Data:

Gibson, John E.
How to do systems analysis / by John E. Gibson, William T. Scherer, and William F. Gibson.
p. cm.
Includes index.
ISBN 978-0-470-00765-5 (cloth)
1. System analysis. I. Scherer, William T. II. Title.
T57.6.G543 2007
658.4′032 – dc22

2006033568

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

Contents

Chapter **10**

The 10 Golden Rules of Systems Analysis

10.1 INTRODUCTION

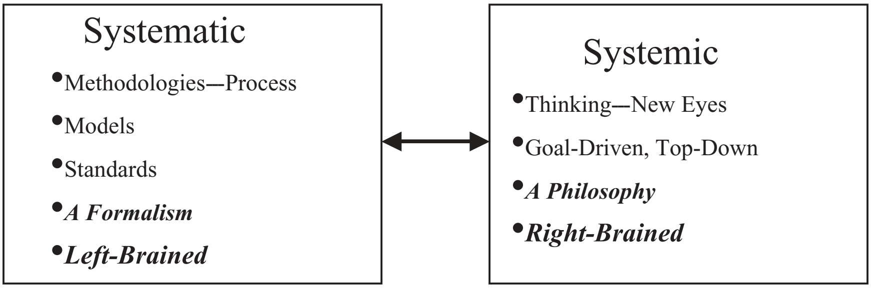
We take it as an axiom that System Analysis (SA) is a practical and an applied activity. We can study mathematics for its intrinsic truth and beauty. But we don't do that in SA. It is fair to say that Operations Research/Management Science (OR/MS) and its close relative SA exist to solve real problems for real clients. No argument to this point. Originally, OR existed to bring mathematics to bear on real problems. We must remember this mathematical core, and without this core of mathematics, SA would have been delayed in its trek into the university world. Nevertheless, if only the mathematics of SA is emphasized, divorced from the client's real-world problem, we lose the whole point. The tools of SA are important, but they are not the product. The product is produced by using the tools of SA for solving real-world problems in the service of real clients. An essential complement to the tools and a component of SA is systems thinking (see Churchman, 1979).

To engage systems thinking, we believe that the analyst needs to see the world with new eyes—that of a "systems perspective." Much of the present literature in the area of systems analysis and systems engineering is very good; however, many sources fail to convey the *art* of systems problem-solving (systems analysis) by focusing instead on either operations research methods (mathematical models such as linear programming) or formal Systems Engineering. Numerous excellent books (examples: Blanchard and Fabrycky, 2006; Blanchard, 2004; Buede, 2000; Dallenbach, 1994; Eisner, 1997; Martin, 1997; Reilly, 1993; Sage and Armstrong, 2000; and Sage, 1992) and handbooks describe the processes of systems engineering, including systems

How to Do Systems Analysis. By John E. Gibson, William T. Scherer, and William F. Gibson

engineering handbooks developed by NASA, DOD, Boeing, and so on (see reference list). Currently, there is also considerable discussion on the concept of system-of-systems (S.O.S.), that is, systems that are of significant complexity and order that they may require methodologies beyond the classic systems methodologies that are all basically derivatives of MIL-499B, a classic systems engineering military standard, which was never actually adopted. The emphasis of this book, however, is not on these formal processes of systems engineering so eloquently described in the referenced books, but on the systems thinking and the associated thought processes.

Fundamentally, we see two worlds typical in systems engineering and analysis—both are necessary:



By systemic, we mean affecting the entire system or holistic.[1] By systematic, we mean a formal step-by-step process (in the most direct form, computer code is an example). An analogy could be made to the left-brain (logical; often engineers) and right-brain (artistic) thinkers. Here, we focus on problem definition and rules we have observed from practice, which is in our opinion a very difficult part of the systems process and an often neglected (or failed) part in practice.

It is important to note that we refer to a very general notion of SA, practiced in wide-ranging domains including health care, finance, policy analysis, transportation, nonprofits, and, of course, all government levels and defense industries.

Thus, we present 10 Golden Rules for Systems Analysts.

10.2 RULE 1: THERE ALWAYS IS A CLIENT

In a sense, we suppose that the first rule is warning you that it is easy to get subjective—that is, to begin fooling yourself. You begin to carve the system problem to fit your tools and to suit yourself, rather than confronting nasty reality. Abraham Maslow, one of the towering figures in American psychology, denigrates some scientists' fixation with the tools of their work (Maslow, 1969). He argues that "technique fixation" is a case of arrested psychological development. Scientific maturity requires problem-focusing, not tool-focusing.

We need a client focus to provide a reality check, to prevent ourselves from becoming subjective and taking the easy way out and also because it is the most efficient and cost-effective way of accomplishing system project goals.

Example. An analyst we were working with on an FFRDC (Federally Funded Research and Development Center) simulation modeling project said "I don't have a client, only a sponsor." In this case, the model is doomed—nobody is interested in the project or the analyst doesn't know whom it is—either way, the project is doomed to inconsequentiality without client input (see Rules 2 and 3).

10.3 RULE 2: YOUR CLIENT DOES NOT UNDERSTAND HIS OWN PROBLEM

Of all the guidelines, this one is perhaps the most difficult for the novice systems analyst to accept. The other rules are interesting perhaps, and some are fairly obvious on the surface, but this one apparently contradicts common sense. But we have heard veteran consultants put it even more strongly. A very successful practitioner once said to us on this subject:

> *"The client always lies. Sometimes he does it deliberately and sometimes apparently inadvertently, but you must never trust the client."*

Now that is what we call strong language. Let's try to see what this accomplished professional was trying to say. Why would a client deliberately lie? In the first place, we have to be clear about what we mean by client. Sometimes a single individual plays all of the roles involved in system analysis, but more often than not there exists some role differentiation. That is, there are a number of personalities involved in the problem solution, and the analyst must attempt to keep their roles clear. We will define and discuss these various roles in detail in Rule #10. For now, we will say that the person with whom you interact is the client, although he or she may not be the decision-maker.

Here is a simple and common example in which the client might deliberately lie. A problem exists: Perhaps an airline is having trouble with its baggage handling system, or perhaps a particular subsystem in a future space mission appears to be behind schedule, or maybe one group within a federal agency wants to let a contract for a particular design study out for bid while another group within the same agency opposes the idea.

Suppose now that a fairly high-level manager in the client organization learns about the problem and brings in an outside organization, of which you are a part, as a systems consultant to suggest a solution. Your boss maybe talks with the high-level manager, but you don't. Then your boss calls you in and asks you to go over and talk to the individual that the high-level manager of the client organization appoints as liaison. Is this unusual? No, we suggest that it is almost universal. We think you will agree that you are going to get a version of the problem that the liaison individual wants you to have. To you, the liaison individual is "the client." He or she is the only person in the whole organization you know.

Suppose that "client" is part of the problem and knows it. Is this unusual? Again, "no." It would only be human nature for that client to attempt to cover up his mistakes, out of fear for his job, from embarrassment, or out of misplaced pride. It would not be at all unusual for the client to lie deliberately and if you take the lie as the truth, you are in trouble.

The client lies accidentally or unconsciously even more often. In fact, one might say that the case of accidental or unconscious lying represents the totality of the remaining situations after deliberate lying is subtracted. This may be overstating the case, but it conveys the principle. We are dealing here with human nature, and there are always exceptions to these rules. It would be naïve of you to think that we can give you scientific certainty.

Think about the situation this way. There exists a problem in the mind of the client, and a system analyst is summoned. Obviously, the client has made an error in reality or in perception. Something is wrong. If the client knew what it was, he would fix it without seeking the analyst's advice. Thus, wouldn't it be dangerous for the analyst to accept the problem definition from someone who has admitted he/she is in error?

The physician listens to the patient recite his symptoms. But she does not accept the patient's diagnosis of the illness. The physician doesn't let the patient decide which are important symptoms and which are trivial. The physician doesn't rely solely on the patient's story, but rather goes on to gather data, take measurements, and order tests. The system analyst should be in a similar professional relationship with the client.

Finally, we note that this is a self-healing counsel. Suppose the analyst takes our advice and assumes that the initial understanding of the problem conveyed by the client is incorrect. Then if the analyst goes on to define the problem as the first step in the system analysis and it turns out that the initial problem statement was satisfactory, no harm is done. On the other hand, if the initial client statement is accepted and turns out to be wrong, the analyst has a failure on his hands.

Example. A consumer client approached us about developing a staffing simulation model in order to optimize the size and allocation of the staff at their chain of retail stores. A quick SA determined that the problem was not staffing but instead store layout and that a simulation model was premature.

10.4 RULE 3: THE ORIGINAL PROBLEM STATEMENT IS TOO SPECIFIC: YOU MUST GENERALIZE THE PROBLEM TO GIVE IT CONTEXTUAL INTEGRITY

You might think this rule is a subset of Rule 2. It is in one sense, but it also is meant to imply something more. Suppose a system analyst starts discussions with a client with full knowledge that the client doesn't understand the problem initially. Through these discussions, the analyst and the client arrive at a mutual understanding of the client's problem. It is that problem which is too specific. The analyst must generalize the client's problem first to arrive at a better understanding of the client's needs, but also to ensure that the problem context will be dealt with properly.

One of the primary ways in which system analysis differs from engineering analysis is the way in which the problem context is handled. One of the basic rules in engineering design is to abstract the problem from its context, in order to do a careful engineering study of the specific issue in isolation. This concept works for a certain class of problem, and indeed the concept of "partitioning" exceeds the realm of conventional engineering design practice. Take the word "interface," for example. The concept of an interface comes from the partitioning process in engineering design. When the engineering designer abstracts his problem from its context, he or she is left with a few, carefully defined "interfaces" or connections between the object to be designed and the rest of the world. There is undeniable strength and power in this engineering approach to design, but not all of physical reality submits to it.

The engineer seeks to optimize the parameters of objects under design in order to meet performance specifications, sometimes called the index of performance, about which we write in Rule 4. The process of engineering design takes a well-defined problem and subdivides it into components to be optimized. This may be termed a bottom-up approach. The bottom-up approach starts with the specific and works up to the general. It is effective and efficient if the problem is well-defined and the method of solution is familiar. It is not as effective in uncharted problem areas and with problems that must be defined by interactions with the client as one proceeds. There, a top-down approach is recommended. This, of course, is the area in which system analysis is practiced.

"Generalize the problem" is all well and good as academic advice, but in practice how do you do it? If you start generalizing with the client, she will likely suspect you of avoiding the problem. Furthermore, it looks expensive to the client if the analyst immediately begins to extend the problem beyond the boundaries of interest. Finally, where do you stop generalizing and start solving? After all, everything affects everything else in the end.

Naturally, there are limits to how far one can generalize. To generalize too far will broaden the topic to the point that it will be difficult to solve it in the time and with the money available. One also runs the risk of alienating the client by appearing to generalize excessively. The client will often be tightly focused on a very specific crisis. If the system analyst takes a relaxed and broad view of the client's problem, the client may object strongly. One must admit furthermore that, in the past, some system analysts have used this generalization dictum to expand the problem beyond all recognition. This usually results in a failure to provide a satisfactory solution on time and within budget. As a result, one often hears arguments against funding another study to produce another report. However:

> A system study that begs the question and has as its major recommendation another study more costly in time and money than the first is a failure. Answer the client's real question on time and within budget.

This is easy to say in theory, but it is difficult to do in practice. The notion of generalizing the question in order to lend it contextual integrity and to define it properly was considered quite bizarre when it was introduced (Gibson, 1973, 1977). It is diametrically opposed to the usual engineering bottom-up approach to problems which consists of subdividing the overall problem into easily solved elements. Yet it is clear that the top-down approach is rather a standard procedure in non-Western societies. Japanese businessmen, for example, insist on examining the contextual integrity of proposed agreements with American firms, much to the annoyance of American managers anxious to "close the deal."

Example. A large consumer products chain wanted us to mange a trade-study involving selecting an Enterprise Resource Planning (ERP) system. An initial client meeting clearly indicated that no goals were established for the ERP—the scope of the problem was much larger, and the need for an ERP not yet established.

10.5 RULE 4: THE CLIENT DOES NOT UNDERSTAND THE CONCEPT OF THE INDEX OF PERFORMANCE

Let's make sure we understand the index of performance concept ourselves before we move on to discuss our duty to inform the client about it. There is embedded here a very strong hidden assumption, about which many system analysts choose not to inform themselves. The concept of optimization is a bedrock axiom from which SA derives its shape and texture, but which is counterintuitive to almost all of human experience. We systems types go about optimizing things. We maximize profit, or mean time between failures, or minimize cost, or installation time, or whatever.

This is not the way in which the world works. Almost no one optimizes anything in real life. In real life, we try to behave so as to avoid major risks. We seek merely to better things incrementally. Discrete incrementalism is the way the world works—discrete in the sense of very tiny steps from the *status quo*. Using a hill-climbing metaphor, the discrete incrementalist climbs a hill with a bucket over his head. He can see only the slope of the hill within the immediate vicinity of his feet. Obviously the discrete incrementalist, if he or she persists in taking tiny steps in the correct direction, will find the nearest local maximum and stick there. If there is a better hill somewhere else in the parameter space, the discrete incrementalist will never find it.

The performance index (IP) is the tool used for measuring the progress the system is making toward the optimum. We have argued that to optimize a system is a radical idea and contrary to most of the folk wisdom of humankind. Nevertheless, optimization is a new and galvanic idea. Previously, people merely accepted the way things were.

Picture Frederick W. Taylor in a Bethlehem, Pennsylvania, machine shop around the turn of the century. Taylor was a bright young lad of inquiring mind and his father was a friend of the owner of the shop. Taylor was a product of the middle class and parental design had him aimed for Harvard, but he would have none of that. He preferred to work with his hands. Taylor immediately began to question current foundry and machine shop practice, but he received no satisfactory answers. He was told by the grizzled master mechanics to keep his mouth shut and to do as he was told. The way they did things was the way they had always been done.

This was the old apprentice system and Taylor would have none of it. Taylor was making himself into a systems engineer. He asked optimization questions:

"What is the best speed at which to run this lathe?"
"What is the best rate at which to load pig iron onto rail cars?"
"What is the best-sized shovel to use for moving sand, or coal, or ore?"

The conventional answer was to do it the way it had always been done or, at most, to make a small change and note the effect. Taylor, on the other hand, sought the optimum by careful experimentation. The measure of improvement he used was straightforward—productivity. He knew that as he increased machine tool speed, for example, tool wear and breakage would increase but he also knew, contrary to conventional wisdom of the time, that this wasn't the point.

Taylor could see a simple optimization hill in one parameter. He could plot productivity versus tool speed and observe that the slope increases to a certain point and then decreases. Above the optimum speed, the cost of excess tool breakage and wear and part spoilage exceeds the value of the extra production gained. This seems like a perfectly simple idea to us today, hardly worth pointing to as the start of a new science.

But, back then, a mechanic was expected to buy his own tools, so his implicit criterion was to save his tools, not to increase production! Here we have a basic controversy. What is the correct optimization criterion? We had it then in one of the first systems studies and it has continued ever since.

Taylor and the shop owner wanted to increase productivity in order to increase profits. The machinists wanted to minimize their out-of-pocket costs for tools, and furthermore they wanted to "preserve the work." The workers knew if the day came that no more work was available, they would be laid off. Note how deeply discrete incrementalism penetrates. To slow down will preserve the work, on this contract. In the long run, of course, the policy is destructive to job security.

One can't count on shop managers to understand what is going on either, as Taylor found throughout his career. Managers have an implicit picture of a well-run shop. It is clean, neat, and orderly. People work quietly and steadily. No running and shouting, no workers sitting idle, no tools breaking and no emergencies. Just smooth, steady production. The implicit IP of these managers is a shop that is easy to manage, not one that maximizes productivity. But Taylor's shops weren't always like that. His index was productivity, and the shops he managed didn't always look well-ordered.

What we have been trying to portray here in this small, isolated anecdote are a few of the differences between the systems approach and discrete incrementalism. We have already noted several facts. First, in the systems approach there must be a distinct, quantifiable measure of performance. In Taylor's shop practice, it was productivity. Second, we saw a refusal by various stakeholders to accept the proposed index of performance. Third, we saw that often the acceptance of the index of performance may require changes in the conventional way of doing things. In shops managed by Taylor, workers did not have to supply their own tools. On the other hand, they did not have the right to go about their jobs in the way they felt best. They did it Taylor's way or they were fired.

Now we are grappling with the fourth rule. As a system analyst, you are in the business of optimization. First, you obtain from the client the goals of the system. This may sound easy enough, but it is perhaps your most difficult task. Then, you obtain from the client an agreement on the measure of performance. No longer can the criterion be left implicit. It must be explicit and measurable. The client usually thinks he understands the goal of the system; that is, he thinks he knows what he wants, although the second rule says he doesn't. The client usually does not think he understands the concept of the index of performance, and in this the fourth rule says he is correct.

We system types understand that one must quantify the IP so that we can rank order the effect produced by each trial solution or "scenario," and thus be able to recommend the optimum to the client. This wasn't necessary in the good old days of discrete incrementalism. Then, the decision-maker needed merely say that a given change would likely produce an improvement over the *status quo*. Political science has never felt the need unfailingly to connect a means of measurement and a quantitative index to every policy recommendation.

Take the legal profession as an example. Is there any evidence available of the effect of jail sentence length on subsequent behavior of criminals? Would a lawyer or judge think to request such evidence? No, of course not. So we regularly incarcerate individuals at an annual cost *per capita* of more than the annual tuition at Harvard, without really knowing or even caring about the marginal impact on behavior? Yes.

All in all, optimization and its measuring stick, the index of performance, provide major conceptual barriers between the system analyst and the client. In addition to the conceptual barrier, there remains the problem of making meaningful measurements.

Critics of SA, such as Hoos for example, make much of the fact that indices chosen are not always directly linked to real objectives (Hoos, 1972). Rather, criteria are sometimes used because they are easy to observe. This is a valid objection. Hamilton et al. (1969) give an example of this fallacy. In a systems study of the recreation potential of the Delaware River basin, the systems team felt it wise to include an index of the purity of the river water. A perfectly acceptable part of a reasonable performance criterion, one would have to agree. The team found that sewage engineers often use the percent of dissolved oxygen (DO) in water as representative of its condition. Thus, the team used the DO content in the Delaware River as the index for recreational potential of the basin. Hamilton argues, and one would tend to agree, that this simplistic criterion hardly represents adequately the overall recreation potential of the entire basin.

On the other hand, this should not be turned into a contest. The systems analyst should not say to the policymaker, "I can measure anything you pick as your goal." This simply isn't true. The analyst and the policymaker should agree in advance on a criterion that is both meaningful and measurable.

Example. A call center that we were consulting for was focused on measuring performance of its call representatives by a measure RPC (right party connect). Again, a quick goal definition exercise and early iteration SA showed that RPC was not related to their main objective—making a profit.

10.6 RULE 5: YOU ARE THE ANALYST, NOT THE DECISION-MAKER

There seems to be a fairly standard cycle of emotions through which system analysts put themselves as they deal with an assignment. The first is fear. It seems only natural for a normal person to approach a system problem with nervousness and fear. After all, the client organization is often well known and thought to be competent, and the specific individual client(s) is/are well-paid and experienced people. If they are having trouble, maybe the analyst won't be able to help either.

As the first meeting progresses, the issue often looks more and more complex. Sometimes, it is apparent from the very first that the client has an axe to grind or has a desired answer in mind. But confusion generally reigns. After the client meeting, the system team talks things over and various ideas emerge. Seldom, if ever, is there agreement at this point as to the real problem. If an apparent, well-defined problem emerges too quickly, analysts should suspect that it is not the real problem. See the Second Rule.

Gradually though, sometimes after much effort, the issue does clarify itself in the minds of the analysts. Now different emotions often take hold; they are arrogance and disdain. How could the client be so stupid as to get himself in this scrape? If he had only done thus-and-so, or avoided such-and-such, this mess would never have happened. And "thus-and-so" or "such-and-such" usually are perfectly standard prescriptions of good management techniques or good engineering design or even just common sense. Notice the implication here that the analyst will *not* have to use advanced or esoteric techniques to solve the typical client's problem.

If an inexperienced analyst is permitted to make the next client presentation, it is almost inevitable that he or she will let the client see this developing disdainful attitude and feelings will get hurt. The client will be hurt by the attitude of disdain, the SA enterprise will be hurt when it is disengaged, and the young analyst will be hurt when he or she is demoted.

It seems almost impossible for the young analyst to overcome this personality deficiency except through experience. It really is evidence of the young analyst's insecurity working itself out of course, but the client won't pay for that. It seems to be totally unconscious and even if the young analyst is warned about it in advance, he can't seem to stop trying to make mincemeat of the client. Thus:

> The analyst must take care of the client. The analyst isn't there to get the client fired. Save the client's job.

The third psychological stage in the maturation of the system analyst, following fear and then arrogance, is one in which the analyst begins to identify so closely with the client's problem that the analyst decides what must be done and begins to play an advocacy role. Identification with the client or the patient is not at all rare in the healing professions, and there is no reason to be surprised when it happens to us. One of the reasons why SA should be practiced in a team setting is to help cancel out this individual analyst bias and subjectivity. Remember, you are not the decision-maker. It is not your place to decide what to do.

By the same principle, as the system analyst, you must internalize the problem. You must be ready to lead the charge, to take the first test flight or test dive. You must follow through by producing a complete transition scenario as part of the final report. And this transition scenario must be prepared with all of the care you would take if your career rested on the success of the project. You must take care of the client.

Example. An analysis case used by the authors with students illustrates this concept very clearly. Students are asked to perform an analysis on a new Wal-Mart store being built in their town. If a student has a preconceived notion of Wal-Mart (such as "I hate/love Wal-Mart"), then they are incapable of performing an objective analysis.

10.7 RULE 6: MEET THE TIME DEADLINE AND THE COST BUDGET

Sometimes we think that people who are attracted to systems analysis are afraid to face reality. Many students can't wait to get to practice their profession. Education majors want to teach, MBAs and commerce students want to make money, engineers want to build things, and computer scientists want to write computer software, but systems students seem only to want more time and money to do another study.

We don't want to solve the client's real problem; we want to tinker some more. Our focus all too often is not a problem to be solved and a better world for the solution. Our focus is the final report. See the Fifth Rule. Unfortunately, the final report often doesn't reveal the truth, it just explains why we did such an inadequate job thus far in the study and why we need more time and money to do a better job in the proposed follow-on analysis.

One of the reasons that some clients go into hysterics if we tell them we are "generalizing" their problem and taking a "top-down approach" is that they suspect this is a trick to avoid facing hard reality. They conclude that we won't finish the study on time and intend to ask for more time and money.

Generalizing the problem isn't a way of avoiding reality, it is a way of coming to grips with it. We must answer the client's real question, on time and within budget. This presents difficulties of course. Think about it this way. The problem probably is ill-defined or the system analyst wouldn't have been hired in the first place. If the problem were well-defined, the client would have solved it himself or handed it to a specialist for solution. The analyst's first task is properly to define the problem, a job the client probably has muffed. So why shouldn't problem definition alone be worth something? It is really. This is the most difficult systems task. After the problem has been defined properly, the issue space narrows and sharper, more straightforward analytic tools become available.

A system study is never really finished. There is always more to do, more scenarios to be produced, more data to be collected, more things to look into, and so on. But the analyst who asks for more time and money to follow up on these will be giving the impression of incompleteness, unless he or she is very careful. We recommend that the decision-maker always refuses to provide more time and money, except under the following carefully defined situations.

As part of the system study, one must always provide a sensitivity analysis and a critical incident analysis. A sensitivity analysis is an examination of the effect on the solution, as system parameters are varied about their assumed set points. Typically, one finds that small variations in most of the system parameters do not affect the outcome significantly. In fact, the analyst should deliberately search for and recommend such "robust" solutions. However, on occasion, it is necessary to recommend to the client a solution containing certain parameters that do have a major impact on the IP. If this is the case, the analyst should recommend to the client a more careful evaluation of these (few and specific) sensitive parameters.

A critical incident analysis is also part of the validation step in a well-conducted SA. The recommended solution for any problem contains many assumptions—for example, the set points of the various system parameters. The SA also contains assumptions concerning the problem environment. The critical incident analysis should contain an evaluation of the effect of any major "off-center" incidents in the problem environment. For example, most urban mass transit studies in the past 20 years assume federal cost-sharing of construction. A change in the percent of this cost sharing would certainly be a critical incident. This whole area is now becoming more prominent and is coming to be called Risk Assessment and Risk Management.

The final report should discuss the major critical incidents ranked by the intensity of the impact. Intensity is the product of the cost of the impact and the probability of its occurrence. If the intensity of one or two of these critical incidents looms very large, the analyst should recommend an additional effort to tie down these estimates more firmly. Of course, if many of the critical incidents have a major potential impact, or if there is a high sensitivity of the outcome to many parameters, the solution is very fragile and should only be recommended in those terms.

So, if we recommend that more time and money be spent, we must be extremely careful to focus very tightly on the way the extra resources will be used and the expected value of the resulting information. This focusing is essential if the analysis is to have credibility with the client.

Example. The concept of ignoring critical incidents, often missed in sensitivity analysis, is typical in many SA activities. Consider the loss of the space shuttle Columbia: Sensitivity analysis was beyond the scope of the foam impact analysis models. However, consideration of a critical incident of a piece outside the scope of the model was warranted, but not considered.

10.8 RULE 7: TAKE A GOAL-CENTERED APPROACH TO THE PROBLEM, NOT A TECHNOLOGY-CENTERED OR CHRONOLOGICAL APPROACH

Obviously, the proper place to start solving a systems problem is at the beginning, right? Wrong. The correct place to start is at the end. We call the step-by-step approach of problem-solving "the chronological approach" because in it we ask, "What happens first?" and we analyze that step. Then we ask, "What happens second?" and so on. We suppose one could say that this is discrete incrementalism applied to the solution process. The chronological approach seems so logical and straightforward that it might be called part of the folk wisdom of humankind. One might also note in passing that if the system approach to problem-solving simply were a collection of folk wisdom, it would not be novel or worth talking about. The problems with the chronological approach are two in the main. The first problem is that it requires the exploration of many blind alleys.

The goal-centered approach appears to be wasting time at the beginning, and it frustrates its practitioners by preventing the release of tension and anxiety to get going. You remember as a kid getting ready to make a family trip in the car. Mom and Dad were trying to remember if they had packed everything and you were jumping around saying, "Let's go! Let's go!" Engineers are like that. They want to start producing computer code without flow charting, and they want to start designing before they understand the problem.

> "Let's doodle around here on the board a bit, just to get the juices flowing."
> "Let's just run a few rough calculations to see what we've got here. Just some back-of-the-envelope calculations, O.K.?"

Not O.K. These early scribblings have a way of committing the group prematurely. Usually, early ideas are conventional solutions and they unconsciously freeze out unconventional solutions. The options field gets narrowed without anyone really knowing it. Note that we are making a strong claim here. We are going further than saying merely that a follow-up to trial-and-error technical suggestions is a waste of time and money. We are saying:

> The technological approach tends to produce an artificially narrow options field and may result in exclusion of superior solutions.

That is the second problem with the chronological approach.

We might want to ask the following question here. Why does conventional wisdom suggest the incremental approach, if it is so apparent that it won't work? We think the answer is contained in the phrase in a preceding paragraph, "Choices keep presenting themselves and without something to go on . . . ." Folk wisdom tells us how the human tribe has handled similar problems previously, so we do have "something to go on." But trial and error is simply too expensive when we confront a new problem, in which we have nothing previous from which to proceed.

Example. In designing a Traffic Management Center (TMC), our clients took a starting position of a computer platform choice. Considerable effort was needed to get them to first consider the goals and measures and then consider alternatives.

10.9 RULE 8: NONUSERS MUST BE CONSIDERED IN THE ANALYSIS AND IN THE FINAL RECOMMENDATIONS

"Beggar thy neighbor" is not a satisfactory system design philosophy. Of course, so boldly put, no one would agree that this is part of her design value system, but by intention or not, it is a design value often used. On the other hand, one would suppose that it is not part of a normal system design requirement to leave the nonuser better off than before, but we will see that this is sometimes precisely the demand.

Let's start with an easy one. How do you feel about graveyards? Many modern middle-class agnostics in America feel that graveyard ostentationism is quite tasteless, and even nominal believers grow uncomfortable at the huge sums of money sometimes spent on necrology in past eras, and even today in some areas. So let's say you are a city planner with a responsibility for urban freeway location. You are aware of the tendency to ram freeways downtown through poor and rundown neighborhoods. Highway planners don't want to discriminate racially; it's just a matter of economics and property values.

But, suppose you have another option. What do you think about running the freeway through cemeteries? Naturally, the graves would be relocated, and so on. Would there be any major objections do you think? Freeway planners in Baltimore and other American cities found to their surprise that the outcry was loud and continuous. Could this have been anticipated had these planners not been so egocentric? Yes, we think so. They could have found out, perhaps by asking the people themselves, or by consulting anthropologists or sociologists.

What's the point of all this? To make our readers uncomfortable? Perhaps. But our conscious purpose is to suggest that:

> The system analyst should respect the value system of non-users and not project his/her value system on others.

Please don't try to decide for the citizens matters over which they should have a say. Don't do public system planning in private. See the Tenth Rule.

Now we'll turn to the other side of the matter. We have to recognize that nonusers are sometimes unreasonable. Many anti-nuclear power protesters are unreasonable, although it must be said that the pro-nuclear fraternity brought the problem on itself. There is no reason why nuclear power plants can't be designed and operated to be among the safest man-made objects in the universe. That they have not been so designed and operated is due to the ignorance and carelessness of the Nuclear Regulatory Commission and utility executives, as well as the original designers. We engineering professors also must bear part of the burden and shame as well. We failed in our obligation of training young nuclear engineers in their public duty and we failed to call attention to potential problems before they became real ones. Our failure is all the more serious, because we have been given tenure so that we could speak out without fear on such matters.

So damage not the nonuser, but let's try to be reasonable, O.K. guys?

Example. The Snail Darter and its impediment on the Tennessee Valley Authority is well-known and a classic, if not extreme, example of a non-user. Another example involves a grocery store in Washington, D.C., that added an oppressive security system to stop theft. Neighborhood residents, who were not considered, were offended and picketed the store, resulting in a shutdown.

10.10 RULE 9: THE UNIVERSAL COMPUTER MODEL IS A FANTASY

You can't get out of a computer model any more than you put into it. This doesn't mean, however, that you can't learn some surprising things from a computer model. We have here the same situation as we have with any mathematical relationship. By definition, all of the results or "solutions" of a mathematical relationship are implied by the relationship itself. But, that doesn't mean you know all of the solutions when you write the mathematical equation. The same is true of a computer model, which, after all, is simply an electronic way of writing equations. What does the mathematics care, whether you write it in ink on paper or write it with electrons in a computer? But it surprises us how few observers recognize this rather obvious fact about computer models.

Of course, the computer can plot very rapidly the solution trajectory for any desired parameter settings and any input and initial conditions. And therein lies its power. But this power is deceiving. It persuades the uninitiated that the computer is a very bright machine indeed. They imagine that one can ask the "giant brain" questions on any subject and get answers, which is nonsense. One can query the "giant brain" only on a very limited set of predetermined and preprogrammed matters and over a very limited range of prestored data values.

The curse of dimensionality is always present in large computer models. What is a reasonable level of detail for the computer to handle for one geographic cell is difficult for 100 cells, and it may be almost impossible for 10,000 cells. The system modeler is usually attracted to the generalized sweep of the model, but she should remember that the client will test the model by specific, individual questions. To provide one more level of detail does not merely add a few more lines of code it won't merely double the size of the program, it may increase its size by a factor of 1,000 or more.

Example. Typical of this quandary is the development of military simulations. It is often the goal to develop the all-encompassing computer-based simulation system; however, repeated attempts have shown that an integrated system of multiple simulation types (hardware and software), humans in the loop, and actual hardware is necessary.

10.11 RULE 10: THE ROLE OF DECISION-MAKER IN PUBLIC SYSTEMS IS OFTEN A CONFUSED ONE

We have referred informally to several roles played by individuals and groups in system analysis. We have mentioned the analyst and his team, the client, and the sponsor. But there are others involved, such as the stakeholders, promoters, opponents, initiators, and advisors, and some authors even mention ghosts. The Tenth Rule says that individuals confuse their roles, thus complicating life for the analyst. This role confusion extends to the analyst himself and the Fifth Rule addresses that point. The Rule explicitly mentions public SA, but corporate and military SA also suffer from this problem.

Let's see if we can define the major role types involved in a typical SA. It is true that role titles are sometimes used interchangeably. Yet each has a slightly different meaning, and so one loses precision and introduces ambiguity by failing to maintain the inherent distinctions.

- <u>The Analyst</u>, or "systems analyst," is the professional who aids those involved with a problem to order the alternatives and decide on a course of action. The analyst should be neutral and objective and should have no stake in the outcome. Yet this counsel of perfection sucks the life and juice from reality. The analyst cannot surmount his own value system and would thus be best advised to declare it to himself and the client when it seems to intrude.
- <u>The Client</u> is a person or group with whom the analyst interacts during goal definition. The client usually receives the final report and sometimes pays the bill. In all probability, the client is affected by the outcome and is thus a stakeholder. The client may also be empowered to choose from among the options offered by the analyst and is thus the decision-maker, but not always. A distinction between client and decision-maker occurs if the analyst is told to deal with the client but the client is not empowered to make the final decision concerning goals and options.
- <u>Stakeholders</u> are all those affected by the system. They may be users or non-users. They need not be clients or decision-makers. Stakeholders may be major or minor, and the ways in which they interact with a large-scale system are myriad.
- <u>The Sponsor</u> pays the bill. The sponsor is usually the decision-maker and may or may not be the client. The sponsor is usually a stakeholder.
- <u>The Decision-Maker</u> chooses from among the options. Thus, the decision-maker must choose the particular index of performance.

One of the most common problems in SA, we find, is figuring out who is whom. Quite often, even before the contract is awarded perhaps, the analyst may be asked to give an initial presentation to a room full of people, under tight time constraints. It isn't always easy to find the sponsor or the decision-makers. We *have* observed that they don't usually sit at the front of the room and they usually don't ask questions. Thus, we suggest you don't get too involved with the vocal critic in the front row, to the point of neglecting others more diffident. However, someone who interrupts your presentation or cuts you off (sob!) is probably revealing his or her power status.

There are almost always ghost actors and hidden agendas. This has been true in all of the system studies which we have been associated with during our careers.

Example. Recent construction of a parking complex at our University brought this issue out—the new garage involved the city, the county, University officials, VDOT (Virginia Department of Transportation), citizens, employees, business owners, and so on, and at many of the decision points it was not clear what group was in what role, resulting in legal challenges, protests, and so on (Rule 8 also came into play: A gravesite was discovered on the parking complex location).

In our experiences, these rules have been invaluable as guidelines for the practice of systems analysis and the teaching of students in systems engineering. We believe that an outstanding systems engineering process, without keeping in mind these rules, is at risk.

Thus, it is done.