A Case Study in Model Management: The US Coast Guard, GAO, TEF A-2, and Structured Modeling with Embedded Languages*

Hua Hua
New Jersey Institute of Technology
Computer and Information Science Department
Newark, NJ 07102-1982
(210) 596-3369
hhua@vienna.njit.edu

Steven O. Kimbrough
University of Pennsylvania
3620 Locust Walk, Suite 1300
Philadelphia, PA 19104-6366
Kimbrough@Wharton.upenn.edu
(215) 898-5133
http://opim.wharton.upenn.edu/~sok/

Abstract

In this paper we present something of a case or field study of model management. We begin by listing and discussing a series of commonly-voiced reasons not to use modeling and model management systems. We then summarize the findings from a GAO (Government Accounting Office) report on a proposal made by the US Coast Guard. The GAO report was highly critical of the analytic work behind the Coast Guard proposal and this led to its rejection. We then revisit the list of reasons not to use models and model management systems and we find them largely implausible. Following this, we describe the elements of a prototype model management system, TEF A-2, which is able to store information about models that is sufficiently rich to generate Structured Modeling reports as well as documented Mathematica files, which can be used to execute the models and obtain graphical output from them. TEF A-2 is a simple system (although it draws upon the sophisticated theory underlying Structured Modeling) and only a prototype. Yet, we believe, it and systems like it can go far to ameliorate the kinds of problems that provoked the GAO report.

1 Introduction

Marginalization and collapse of esteem are terms that certainly overstate the present unhappy status of management science modeling. Nonetheless, recent trends—particularly low job market demand for management science analysts, transfer of operations research research activity into operations management, and ongoing reduction in the academic commitment to operations research—have led some to wonder whether, or to fear that, what is presently an overstatement will soon become no exaggeration at all. The state of management science modeling, thus, might best be described as malaise (see [21] for confirming statistics).

We are building another model management system, called TEF A-2. Our main purpose in this paper is to report on it. This work was sponsored, in part, by the US Coast Guard, and a number of Coast Guard models have been put into TEF A-2. But to appreciate what we are doing, and why, the present malaise in management science modeling must be acknowledged and addressed. Why, if modeling is in decline, should one build a model management system at all? To put the point more positively: let us understand more about the context in which modeling is conducted and we shall be better able to design effective and appropriate model management systems.

We begin, then, with some comments on the use of models to support managerial decision making. We present something of a field study on modeling at the US Coast Guard, believing that its lessons are quite generally applicable. Our comments are informed by, and responsive to, secular trends in management science (noted above), our experience with the US Coast Guard, and the oral tradition in this field (i.e., numerous discussions with various people over a period of several years).

*This research was supported in part by US Coast Guard contract number DTCG39-86-C-E92204, Steven O. Kimbrough, principal investigator. File: hua-sok-tefa-2-hics97-final. 960924.
2 Background on Model Use

Whatever the reason that modeling activity is in, or is threatened with, decline, it is not because everything that can be modeled usefully has been. Indeed, Little’s lament that “The big problem with management science models is that managers practically never use them” [19, p. B-460] was old when he made it 25 years ago.1 It remains valid today, as nearly all observers of the scene agree, including Geoffrion (e.g., [8, 11]) who has perhaps been the most acute and articulate. The question is Why?

A number of reasons, or excuses, have been given for not using, or managing, models in an organization. Those we have most often encountered are listed below, along with brief comments.

1. Our organization doesn’t use models. And, by implication, doesn’t need them. We find it interesting that no one claims that better decisions can be made without models. After all, there is no more robust finding from behavioral science than that humans are rather poor decision makers and can often easily be beaten by rather simple models.2 Rather the claim is (often) that models are somehow irrelevant. We agree that modeling has often been oversold and that its useful realm is not unlimited. Further, computerized exploration of data (with or without graphics) and organizational re-design (business process re-engineering and related enterprises) can often serve as substitutes for, or alternatives to, management science modeling.

2. Managers don’t understand models and so won’t accept them. Little [19] addressed this point straight on. His recommendation, widely recognized as seminal in the field of DSS, was to computerize the models so that analysts could more thoroughly explore them in order to gain understanding of and confidence in the models. Geoffrion [8, 9, 12, 11] is in avid agreement with Little’s recommendation, and adds that following the discipline of proper design and representation of models can only promote their use and value.

3. When we need a model we get it from another organization, or we get someone to build the model, and then we use it. By implication, no systematic approach to managing models is needed. It is appropriate to recognize forthrightly a conflict of interest between an organization and its consultants. The consultants (and their organizations) have an interest in having the customer (purchasing) organization rely on them. To the extent that a model is a black box and cannot be understood or modified or validated independently, then to that extent the organization must continue to rely on its consulting vendor. We take the point of view of the customer organization, whose interest lies in making its models as transparent, as modifiable, and as validateable as possible.

4. The models that we need and use make decision making trivial. The implication is that modeling consequently does not need to be attended to, nor do the models need to be thought of as valuable organizational assets that should be carefully managed. This implication is essentially irrelevant to

---

1Note the ambiguity. Are managers merely failing to use models directly, by using software to interact with them, or is the failure of an indirect sort, with the managers refusing to use results obtained by others? The worry is that it is the latter that is the problem.

2The literature here is enormous. It is often associated with studies by Kahneman and Tversky. Piattelli-Palmarini [20] is a recent review and contains many useful references, omitted here in the interests of saving space, to the standard works in the field, but see [13] for an especially good collection of papers. Dawes [7] is a particularly compelling reference on how simple models can greatly outperform people.
the attendant claim. Even if decision making is easy, once the data have been collected and the models have been run, the models still have to be maintained, understood, and explored. The fact that they are so good at supporting decision making makes such models more valuable and more worth the investment needed to manage them effectively.

5. Our model building is always a one-shot affair. Models are used on a project basis and then abandoned. This may be true, but it is not something to brag about. Projects get audited, decisions get revisited. Without effective management of the models they will be irrecoverable, or recoverable only in an expensive and untimely fashion. Further, abandoning models in this way is to preclude the possibility of using pieces of them later, or even of learning from them. The cost of effectively archiving a model is so low that it is simply not plausible, in most cases, that it is not worth paying.

6. The analyses we need to do require special treatment and this cannot be done with existing modeling and model management systems. Often, this complaint is merely a form of special pleading. Everyone thinks, and every consultant knows, that his or her expertise and know-how is unique and not to be governed by ordinary rules. This said, it has to be admitted that the features and functionality of today's model management systems leave much to be desired. But the same is true of database systems. Database and model management systems have much to offer, and have no requirement to offer everything. As in the case of database systems, the burden of proof should rest with those who prefer not to approach the management of an organization's models in a principled and systematic fashion. Just as the default assumption is that an organization's data should be managed with a well-normalized relational database, so the default assumption should be that an organization's mathematical models should be stored in a principled and systematic model management system.

Some general comments about the above complaints. They are mutually inconsistent. This is surely understandable when different individuals register their views. In fact, however, we have heard individuals espouse such mutually inconsistent complaints still, the complaints have to be answered. How? We offer some evidence. It is indirect. It is anecdotal. Yet we believe it is representative and general.

The following passage constitutes the executive summary ("Results in Brief") of a document issued by the United States General Accounting Office. The document, Adequacy of the Justification for Heritage Patrol Boats, reports on an audit (by the General Accounting Office) of a proposal by the United States Coast Guard to acquire 47 Heritage Class (120-foot) patrol boats.

The Coast Guard's project to acquire Heritage vessels was not adequately justified and did not closely adhere to federal guidance for acquiring major systems. As a result, the acquisition has fallen behind schedule, and the vessels will not be available when needed, raising questions as to whether the Coast Guard will be able to effectively conduct its missions. Specifically, we found the following:

- There were weaknesses in identifying mission needs and the capabilities the re-
placement vessels would require to meet these needs. The Coast Guard also could not support its decision for the number of patrol boats needed because agency officials could not provide support for the calculations of the computer model used to determine the need for 96 vessels.

- The Coast Guard identified different patrol boat designs that would satisfy its needs but selected the Heritage design without completing the competitive process. In addition, the Coast Guard's evaluation did not consider all relevant costs, such as those that would be incurred to renovate over half of the home ports to accommodate a vessel as large as the Heritage boat.

- The Coast Guard's initial estimates understated the time and cost required to obtain Heritage boats and lacked support. Agency officials acknowledged that factors such as underestimating the time needed to design, develop, and produce a prototype have caused the date for the beginning to acquire Heritage boats to slip from 1990, as originally planned, to 1996. Because fleet replacements will not be available when needed, the Coast Guard has had to extend the lives of older patrol boats that it considers to be less than adequate for performing patrol boat missions. The current cost estimate for the Heritage prototype, now under construction, is $13 million, nearly double the initial estimate of $7.7 million.\(^3\) Subsequent production-run vessels are currently estimated to cost about $7 million each, although the Coast Guard cannot document this figure.

The Coast Guard is currently reevaluating the appropriate size and composition of the fleet, assessing the costs of renovating ports, and considering acquiring smaller, less expensive "off-the-shelf" (commercially available) boats in place of some Heritage boats. [22, pp. 1-2]

In short, the GAO found the Coast Guard's proposal woefully lacking.\(^4\) The proposal was rejected. The case before us is, we believe, atypical only in that the documentation of the failure is carefully recorded and made public. What does it teach us?

Without unduly elaborating the point, the case at hand gives the lie to the complaints, listed above, against modeling and model management.

1. **Our organization doesn't use models.** Here is a case in which the organization did use models, and in fact had to use them, there being no genuine alternative to modeling for estimating costs and performance of boats and operating policies.

2. **Managers don't understand models and so won't accept them.** The managers certainly did not understand the models. The GAO is right in saying that the models were not sufficiently documented to be understandable at all. But offered the choice—between (1) using a model management system that could facilitate understanding of and insight into the models, and (2) proceeding with a recommendation not based on substantial understanding of the models—few would defend the latter.

3. **When we need a model we get it from another organization, or we get someone to build the model, and then we use it.** Indeed, some of the key models used to support the Coast Guard's proposal were obtained from other organizations, in particular the US Navy. Failure of the organization to understand these models, and to adapt them for use by the Coast Guard, led to improper use of these models—and to recommendations that were not credibly supported. Especially when models are developed outside an organization, there is need to manage the models, to document them, and to have machine support for validating them and for making their meanings transparent.

4. **The models that we need and use make decision making trivial.** A well-known fact about decision making is that everyone thinks they are experts at it and no one admits to doing it on the job, unless it is retrospectively on an acclaimed success. The GAO report, again, gives the lie to this complaint against modeling and model management. In the present case, even if it could be calculated what various configurations cost and how

\(^3\)Both cost estimates are in constant 1991 dollars.

\(^4\)We are familiar with this case and know of nothing that in any way undermines the GAO's findings. The report is on the mark. If anything, it is understated.
they would perform, the search space of policy options remains enormous. Making tradeoffs is difficult and fraught with political risk, so it is understandable that it is downplayed or hidden. Yet tradeoffs are there, on the table or under it, in any proposal to allocate resources. Requirements for explicit, sound justification can only be expected to become more rigorous in today’s tighter, more competitive environment.

5. Our model building is always a one-shot affair. Models are used on a project basis and then abandoned. Not so in the case at hand. First, it is usually not at all clear when a project ends and the models can be abandoned. Auditors and lawyers may arrive years later, requesting documentation and reconstruction of reasons for a given action or proposal. Throughout the life of a typical project, frequent requests—for documentation, for reconstruction of reasons, for examination of other alternatives—are normal and expected. Answering these requests in a timely and accurate manner can only be furthered by systems for carefully and systematically managing models.

Second, part of what makes the GAO report so devastating is that certain of the models it criticizes are standard models, intended to be used in all vessel acquisitions and planning exercises by the US Coast Guard. The simple fact is that here, and quite generally, although particular decisions are unique, they tend to fall into patterns, for which particular models and decision making methodologies may, often with some modification, be repetitively applied. Understanding the costs and performance capabilities of a 120-foot patrol boat will have much in common with understanding the characteristics of a 110-foot boat, and even a 378-foot ship. To abandon experience and modeling assets is simply to engage in unnecessary waste.

6. The analyses we need to do require special treatment and this cannot be done with existing modeling and model management systems. As noted above, a model management system may be helpful even if it does not do everything needed or useful for managing models. The same is true of a model: it may be useful as part of a solution. What, then, may we reasonably hope for in a model management system, and how would it add value? We turn now to the particular contribution made by one such system, TEFA-2.

(* Section 4: Variable Descriptions *)

bbpF::usage = " bbpF is defined as a local variable for the global variable bbpF. bbpF : the basic Barrier Patrol Function."

l::usage = " l is defined as a local variable for the global variable length . length , in the Basic Barrier Patrol model, is the length of the patrol barrier measured in nautical miles."

r::usage = " r is defined as a local variable for the global variable radius . radius , in the Basic Barrier Patrol model, is the radius of detection of a patrolling Coast Guard vessel . r is measured in nautical miles."

v::usage = " v is defined as a local variable for the global variable speedOfCGPV . speedOfCGPV, in the Basic Barrier Patrol model, is the speed a patrolling Coast Guard vessel . r is measured in knots or nautical miles per hour."

u::usage = " u is defined as a local variable for the global variable speedOfTargetV . speedOfTargetV , in the Basic Barrier Patrol model, is the speed a target vessel . it is measured in knots or nautical miles per hour."

Figure 3: TEFA-2 Mathematica Output (edited to save white space): Section 4 for the Basic Barrier Patrol Model

3 An Application Theory for Model Management Systems

The GAO passage is richly suggestive for what the requirements of a model management system should be. We leave detailed analysis to the reader and to subsequent occasions. Briefly, a model management system is, or is part of, a DSS. The main purpose of a DSS should be construed as providing support for constructing, comparing, and evaluating arguments for alternative courses of action. This is called the argumentation theory of DSS (see [17]). In our present example, the argument in question was a proposal by the US Coast Guard to acquire a new fleet of Heritage Class patrol boats.

Models are appropriate in a DSS when they are useful for constructing, comparing, or evaluating arguments pertinent to the issues at hand. We have argued for the present case, and believe the point applies very
generally, that models are often essential and required for credible argumentation. Proposals for decision and action are often simply not credible without quantitative, analytic support. Management science models are especially appropriate (and required for plausible argumentation) when resource allocation decisions are at hand, as they very often are. The need for optimizing the allocation of scarce and expensive resources in a huge search space is a frequently encountered fact of life, whether or not a particular instance may usefully be formulated as a mathematical programming problem.

And model management is appropriate in a DSS whenever models are. The argumentation theory helps us see why and see what some of the requirements for a model management system are. Briefly, arguments are convincing and credible when they are valid (or at least strong\(^5\)) and their assumptions are credible (or better). Conversely, an argument whose validity or assumptions cannot be ascertained or examined is an argument that few will find palatable. Model management systems, at the very least, should help make models—their use, their workings, their assumptions—maximally transparent. Further, and contributing to the transparency of models, a model management system should provide material assistance in assuring the validity of models and their inputs. Finally for the present—since we are not claiming to have spanned the space of model management requirements, far from it—a model management system should help make models easy to use and to explore rapidly. We note that merely having a common, systematic approach to representing and storing models, a model management system can reduce the cognitive overhead of having to learn a new interface for each separate model encountered.

In sum, management science models are often needed because they are essential elements in making a credible argument for a course of action. Model management systems are needed to reduce the costs and improve the quality of working with models, and they need to bring three important values to the table: transparency, validation, and speedy exploration (of models, and of their assumptions and inputs). The point is perhaps most succinctly captured in the request we heard from a captain, "I want a system that tells me where the numbers came from." [17]. The GAO report is essentially a complaint that it could not discover—at other than a trivial depth of analysis—how the proposed number of new vessels was determined. Models were used to estimate requirements, but the relevant assumptions and many of the important formulas and input values were obscure, not only to the GAO, but to the Coast Guard itself. To the extent that a model management system can help answer questions about modeling results, those results (if revealed to be valid) can only be more credible.

There is much more to be hoped for in model management systems, but the benefits before us are quite substantial. Let us now see how a particular system, TEFA-2, manages to deliver these benefits.

4 TEFA-2: A Brief Description

TEFA-2 is presently dispersed across three code systems: a Unix implementation in Prolog with an XWindows graphical user interface; a Macintosh implementation in Prolog with a tty interface; and a series of WWW (World Wide Web) pages containing reports produced by either the Unix or Macintosh implementation of TEFA-2. The Macintosh version is now the primary development vehicle. After further development, we plan to port the Macintosh system to the Unix environment and attach it (via forms programming and CGI) to the WWW pages, thereby allowing users to issue commands to TEFA-2 over the Internet. The WWW pages may now be accessed on the Internet via: http://opim.wharton.upenn.edu/~sok/tefa2/.

TEFA-2 uses the embedded languages technique internally to represent models and information about models and their inputs [1, 3, 4].\(^6\) As such, TEFA-2

\(^5\)Strong arguments need not be deductively valid, but must command general assent. Inductive arguments and arguments by analogy often have this characteristic.

\(^6\)The details of the representations used in TEFA-2 vary from those used in earlier versions of TEFA.
TEFA-2 has very rich and open capabilities for representing information relevant to models. This information is exploited in TEFA-2 through a growing series of drivers that are able to transform TEFA’s internal representations into formats useful for various purposes. For example, TEFA-2 does not contain any model solvers. Instead, TEFA-2 relies on external solvers, preferably commercially available and widely-accepted solvers, such as Mathematica.

To execute a model represented in TEFA-2, one issues a TEFA-2 command that generates a Mathematica notebook file. This is an ASCII file and can be interpreted by Mathematica, regardless of host machine. An example of a file generated by TEFA-2 in this way is presented in figures 1-5. Several points arise.

1. TEFA-2 currently has a Mathematica driver and can produce files as illustrated in the figures. There is nothing special, from TEFA’s point of view, about Mathematica. Drivers may as easily be provided for other solvers, such as Maple, AMPL, and GAMS, or even spreadsheets.

2. Under the design concept for TEFA-2, a model becomes a (Mathematica) document, and so can be treated and managed as a document. It may be transported, subjected to string searches, archived with other documents, and so on. By being able to generate models as documents, TEFA-2 allows document processing and management technologies to be applied to management science models. For example, models represented in this way could be made subject to search and indexing by the standard Web crawlers. Also, such document models could be processed by any of the rich panoply of retrieval techniques that have been developed and investigated by the Information Retrieval community (for a recent review, see [16]).

Further, an organization may keep (and centralize expertise on) one copy of TEFA-2, but use TEFA-2 to generate the modeling documents, which may then be dispersed throughout the organization. Indeed this is just the intention behind our Web pages for TEFA-2: the user may browse for information, select a particular model, and download it as a (Mathematica, etc.) document for local manipulation and execution.

3. Model documentation is essential for making models transparent, for validating them, and for quickly learning how to use them. TEFA-2 is designed (recall: embedded languages) to represent a rich and extensible body of information about models. This information is available interactively in the Unix and Macintosh systems. More importantly, TEFA-2 has drivers to present the documentary information in other forms and formats. Note, in figures 1-5, that the model is amply documented in a TEFA-2 standard format. Some of this documentation can be understood by Mathematica and when this is the case, TEFA-2 generates appropriate Mathematica code. Otherwise, TEFA-2 generates what to Mathematica are comments, but these are useful to humans and document processing routines (e.g., keyword search engines).

4. Other drivers are available in TEFA-2. Most notably, the internal representation in TEFA-2 is sufficiently rich to capture structured models at levels 1 and 2 [9, 12]. We see in figure 6 the Structured Modeling Language schema for the level 2 model, SATELLITE, described in [10, pp. 3-2 to 3-4] (see also [12]). The contents of figure 6 were generated by TEFA-2, whose internal representation of models and information pertaining to them is quite different. By merely adding this driver, TEFA-2 models can be transformed to SML representation and then checked and validated from the point of view of, and with existing software for, Structured Modeling. Thus, the TEFA-2 model management system avails itself of a considerable and impressive body of work in model management. (We are presently engaged in extending the TEFA-2 SML driver so that all SML data structures, for models at levels 1 through 4, can be generated.)

5. TEFA-2 is capable of further representation and processing for models. The current system contains elementary dimensional checking and validation for models. We are expanding this feature. In addition, we plan to add a module for testing quiddities, which may be thought of as a generalized form of dimension and as a way of detecting further invalidities in models [5]. Our intention is to have TEFA-2 generate validation reports and to make these available via the Web pages, either independently or as part of the (Mathematica, etc.) model documents.

TEFA-2 has other interesting features and capabilities, but these are the most important and innovative, and most to the point for present purposes.
Section 7: Model Evaluations

Print["Model Output:"]
Print["According to the model(s) and the scenario, the variable bbpF[x,v,u,l] is equal to ", bbpF[x,v,u,l]]

(* Section 8: Post-Evaluation Analyses *)
x1 := bbpF[x,v,u,l]
Plot[x1, {l,100,300}]
x2 := bbpF[x,v,u,l]
Plot[x2, {u,15,30}]
x3 := bbpF[x,v,u,l]
Plot[x3, {v,15,50}]
x4 := bbpF[x,v,u,l]
Plot[x4, {r,10,50}]

(* Section 9: End of Input *)

Figure 5: TEFA-2 Mathematica Output (edited to save white space): Sections 7, 8 and 9 for the Basic Barrier Patrol Model

5 Discussion and Conclusion

The features and capabilities of TEFA-2 go far to meet the requirements identified by the GAO report discussed above. In doing so, we believe that these features and capabilities address a quite general set of requirements for management of management science models. We have summarized these requirements as (at least) a demand for transparency, validation, and speedily exploration (of models, and of their assumptions and inputs). Specifically, many assumptions—such as the input variables and their intended meanings and dimensions, the mathematical structure of the models, those responsible for further documentation and maintenance, and much else—are made entirely transparent. How is speed of the Coast Guard vessel represented in the Basic Barrier Patrol model and what are its dimensions and units? This is very easily answered, either directly in the TEFA-2 software itself, or indirectly from the (Mathematica) modeling documents. Many other such questions are as easily and transparently answered using TEFA-2 (directly or indirectly). Moreover, there is considerable benefit in this regard from having a common, standard system (TEFA-2) or environment (Mathematica modeling documents, as in figures 1-5), applicable for many different models. Finally, we note that being able to support a recommendation with a model that is demonstrably valid according to a variety of criteria (e.g., dimensional validity, Structured Modeling validity) can only enhance the credibility of the recommendation and recommends.

None of this is to say that TEFA-2 is the best all-around model management system there is. At present, it certainly is not. (For discussions of other or related modeling systems see [13, 14, 18].) TEFA-2 is an experimental system, developed to explore and test ideas. Our purpose in the present essay has been twofold. First, we have aimed to provide something of a case study—with public, objective documentation—for why good modeling and model management systems are needed. We believe that the model management literature would benefit by having more such cases available, but at least here is one that is generally useful. Second, we have aimed to describe briefly certain of the central ideas and capabilities behind TEFA-2, a model management system that is currently under development. There are three main ideas here that we have sought to convey.

1. Use of the embedded languages technique to extend the original TEFA modeling system [1, 3, 4] to represent Structured Modeling ideas.

We believe that Structured Modeling concepts, developed by Geoffrion, have enormous potential to improve modeling practice. We are Structured Modeling supporters. On the other hand, we do not believe that the information captured by Structured Modeling is complete for all modeling and model management purposes. The significance of using embedded languages to represent Structured Modeling information is that the embedded languages technique is extremely general and can represent much other useful information (e.g., [5]) as well. We have now demonstrated that it works for Structured Modeling and we can report that it works well, although a much longer paper will be required to fully substantiate this. If we are right about embedded languages, however, we have before us the happy prospect of being in possession of a general method for representing various different, but useful theories for model management.

Footnote:

For example, SM does not offer a fully formal representation and means of handling certain semantic information pertaining to modeling, such as dimensional information (cf., [2, 5]). Nor, e.g., does it support exotic, but possibly useful, forms of logical inference about modeling (cf., [6]). This is not, or at least not much of, a criticism of Structured Modeling. What it does it does well and SM can be extended. We are grateful for that.
2. Use of (textual) documents to represent models and information about models.

TEFA-2 generates executable Mathematica workbooks, with extensive comments and documentation built in. These notebook files may be used on any machine on which Mathematica executes. Moreover, they are laid out in a clear, understandable, and regeneratable fashion. What if the models used in the Coast Guard’s recommendations had been implemented in this way? Further, as noted above, this strategy allows users to treat model instances as documents, and to store and retrieve them in familiar ways.

3. Use of a simple system that provides substantial help in addressing the known problems of modeling in organizations.

TEFA-2 uses embedded languages to represent models and information about them. The present implementation requires a Prolog programmer to implement a model, but once a model is implemented, various drivers are available to the user in order to extract, e.g., Structured Modeling information, and executable Mathematica workbooks. The principle generalizes: given a rich representation format, drivers for needed functionality can be added incrementally and (our experience) fairly easily.

In conclusion, we return to the malaise of contemporary management science. All great cities have their times of ascendancy and descent, yet they endure and even in their nadirs remain impressive centers of activity and influence. So it is, too, with great intellectual disciplines. The workings of supply and demand are dynamic. We should attend to the fundamentals, and these are solid. Briefly, we have pointed out that models will inevitably be used in decision making. For many sorts of important decisions, management science models are the only game in town for constructing a convincing argument for a course of action. Even if the more exotic and abstruse types of models fall into desuetude, there is ample opportunity to add value by supporting the use of simple cost, forecasting, and performance models. And given well-designed model management systems, the incremental cost of working with other models can only be reduced. Effective model management systems, sorely needed in today’s environment, will increase the market’s demand for modeling. These facts, and the continuing advancements in computer and communications hardware and software, constitute a convincing argument for optimism.

&SATCHELLITE
SATCHELLITE /pe/ There is a SATCHELLITE in space.
OBJECT /pe/ There is an OBJECT in space.
S_MASS ( SATCHELLITE ) /a/ : Real+ The SATCHELLITE has a certain SATCHELLITE MASS in kg.
O_MASS ( OBJECT ) /va/ : Real+ The OBJECT has a certain OBJECT MASS in kg.
D ( SATCHELLITE OBJECT ) /va/ : Real+ THE SATCHELLITE and OBJECT are a certain DISTANCE apart in meters.
FORCE ( S_MASS O_MASS D ) /f/ ; 6.67 * 10^(-11) * S_MASS * O_MASS/ D^2 The OBJECT exerts a certain FORCE on the SATCHELLITE, in newtons, according to Newton’s Law of Gravitation.
THREAT ( FORCE ) /t/ ; FORCE > 10^(-6) The OBJECT is a THREAT to the SATCHELLITE if and only if it exerts a FORCE of greater than one millionth of a newton.

Figure 6: Structured Modeling Schema Generated by TEFA-2: Level 2 SML Schema for the SATCHELLITE Model (edited to save white space)

References


