PROJECT TRADES MODEL FOR COMPLEX SPACE MISSIONS

André R. Girerd and Robert Shishko, Ph.D.*
Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109

ABSTRACT

A *Project Trades Model* (PTM) is a collection of tools/simulations linked together to rapidly perform integrated system trade studies of performance, cost, risk, and mission effectiveness. An operating PTM captures the interactions between various targeted systems and subsystems through an exchange of computed variables of the constituent models. Selection and implementation of the order, method of interaction, model type, and envisioned operation of the ensemble of tools represents the key system engineering challenge of the approach. This paper describes an approach to building a PTM and using it to perform top-level system trades for a complex space mission. In particular, the PTM discussed here is for a future Mars mission involving a large rover.

INTRODUCTION

The requirement for tradeoff studies is an established tenet of systems engineering practice. In order to allow the selection of the best course of action, tradeoff analyses provide decision makers with recommendations, predictions of the results of alternative decisions, and appropriate supporting information.[1] In order to perform the analysis in a rational, objective, and repeatable manner, there must be an agreed-upon approach for measuring alternatives against criteria.[2] Organizing the available data and resources to support an effective evaluation under suitable criteria becomes a key systems engineering challenge, quintessentially important early in a project's life. The necessity of a well-structured approach is all the more pressing for complex space missions, where the number of ripple effects across diverse subsystems overwhelms intuitive system understanding, and design definition choices impart significant cost, performance, risk, and schedule ramifications.

Complex space missions with multiple interactions between systems and subsystems are particularly suited for representation by a network of interlinked models. We use the term *Project Trades Model* (PTM) to describe this collection of performance, cost, risk, and mission effectiveness models. The construction and use of a PTM is an effective and powerful approach to enable tradeoff analyses. As a systems model, it provides the opportunity to see a system from multiple perspectives or several viewpoints at the same time. It thus enhances our ability to comprehend system behavior and to under-

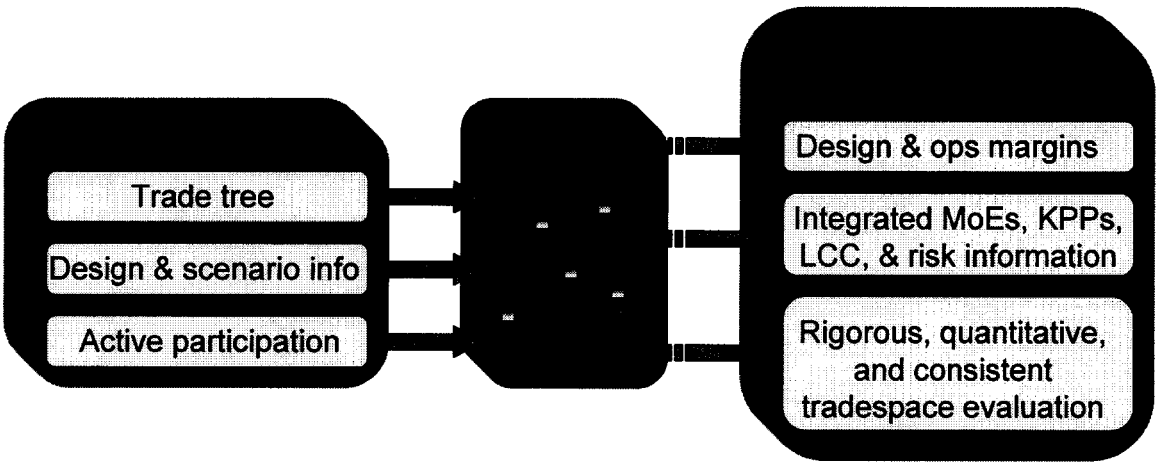

Figure 1 — PTM Inputs and Outputs

* System Engineer and Principal System Engineer/Economist, respectively

stand the important interrelationships, improving clarity and insight.[3]

This paper describes an approach to building a PTM and using it to perform top-level system trades for a complex space mission. In particular, the PTM discussed here is for a future Mars mission involving a large rover.

At its core, a PTM is an Analysis of Alternatives (AoA) method that seeks to weave together "the threads of calculation" necessary to quantify mission-level Measures of Effectiveness (MoEs), system Key Performance Parameters (KPPs), and project Life Cycle Cost (LCC). These calculations are implemented by integrating appropriate space system design, cost, and operations models/simulations. Figure 1 shows the PTM in context with its output deliverables and required project inputs. In order to calculate the specific desired output metrics (MoEs, KPPs, and LCC), and capture the distinctive and complicated system and subsystem ripple effects of the targeted mission, each PTM must generally be custom-made, though the reuse of components for future similar missions is fully expected.

Although geared toward top-level trades, the expectation of reaching a perceived engineering credibility threshold drove the large Mars rover PTM to a level of detail significantly beyond the simple and toward the use of existing, "best practice" legacy tools across the Jet Propulsion Laboratory (JPL). The experimental component of this PTM effort was to integrate these tools into a distributed system using Web Services. In doing so, we aimed to utilize recent advances in Internet technology to solve problems that befell earlier generations of PTMs, which were not distributed.

An emphasis on (life-cycle) cost-effectiveness necessitated a focus on operational performance during the surface mission. To accomplish this for an in-situ mission with multiple science objectives and a complex (and indeed, uncertain) physical operating environment required a clear interconnect between the system design and the operational scenario. Further, these important and unavoidable uncertainties (in operations and costs) resulted in the use of Monte Carlo techniques to estimate distributions rather than treating these phenomema as fully deterministic.

General Benefits of the PTM Approach

The PTM approach offers a number of advantages over traditional methods of tradespace exploration. In the traditional approach to mission definition and early design, engineers use stand-alone models to assist downselect decisions of system or subsystem options. (See Figure 2.) This approach is least objectionable when the individual tradeoffs can be safely decoupled from each other. However, the trade choices still arrive sequentially in time, risk inconsistency from changing assumptions during the interstitial period, and provide the end result of a single system description or point design. The prohibitive time required to repeat the cycle places a practical bound on the number of system designs for evaluation.

The PTM approach invests time upfront identifying systemic interactions and assembling models to capture subsystem couplings. When constructed, the PTM offers rapid generation of candidate designs, resulting in a sufficiently populated tradespace to enable the emergence of a frontier. Any assumptions

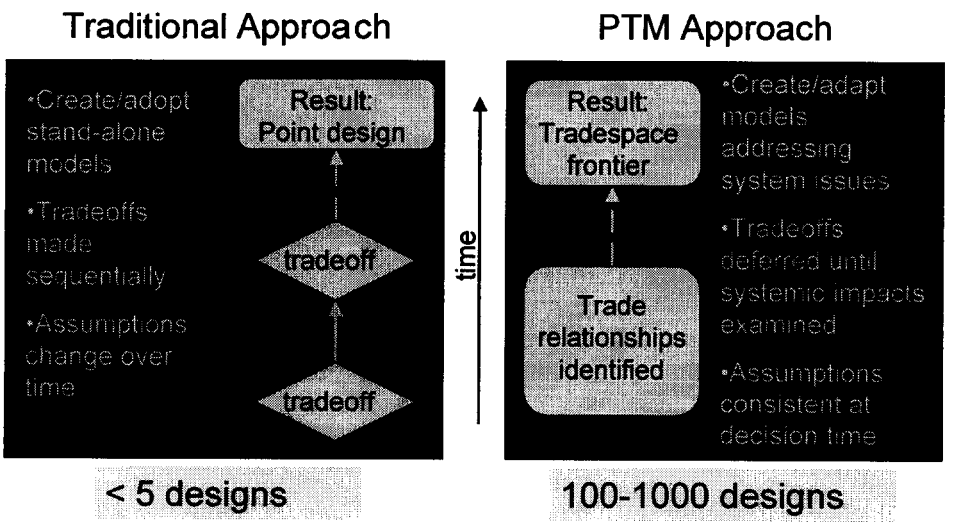

Figure 2 — Old Problem, New Approach

are consistent at decision time, and may be changed across all models as the need arises. Furthermore, the PTM automatically captures design inputs, outputs, and rationale, assisting traceability of decisions.

Model-based trade studies represent an increasingly popular design approach in industry and academia. At the Massachusetts Institute of Technology, for example, aerospace graduate students can take Course 16.89 (Space Systems Engineering), where they design, construct, and integrate models to analyze alternatives for chosen space missions. Past projects have included the *Terrestrial Planet Finder*, an ionospheric observing swarm, and a Mars vehicle. The methodology and results typically garner enthusiastic receptions from industry observers.

A primary benefit of the PTM approach is a reduction in the marginal cost and time of analyzing new alternatives, which allows for the creation of a sizeable and consistent output dataset that can be mined for optimal solutions. The ability to make this output dataset the solution-space of a broad range of candidate design parameters is particularly useful early in a project's life, when the tradespace should be comparatively open and a baseline has yet to be adopted.

Drawbacks of Previous PTMs at JPL

However, as the number of highly dissimilar design options increases along a growing number of different trade axes, the dimensionality of the modeling problem expands dramatically, causing considerable implementation difficulty. To combat this, PTMs geared for broad explorations tend to be composed of low-to-medium fidelity models. (For this class of problem, universities have shown particular aptitude.) The problem is made tractable by a consistent application of moderate scope and by integrating the models for system-wide studies onto a single computational platform. Examples of such previous efforts include the custom software-based System Design Tradeoff Model for Space Station Freedom (1985-1990) and fully Excel-based PTMs for the Pluto Fast Flyby, Mars '98 Orbiter, SIRTF, and Europa Orbiter missions (1994-2000).

Unfortunately, this approach experienced a number of drawbacks:

- The awkwardness of passing around the integrated model (usually in one workbook) to potential contributors put a damper on improvements and team inclusion. Typically, discipline experts maintained and used their own models and tools in a stand-alone mode.
- Spreadsheet-based approaches were viewed as "lightweight" with low fidelity.

The combined effect of these drawbacks hampered the unambiguously successful adoption of past PTM efforts despite the added value demonstrated to their parent projects.

When engineering credibility becomes the driving factor, implementation complexity increases, even when accompanied by a narrowing of breadth. As the desired trade focus shifts from wide and shallow to narrow and deep, the fidelity of the constituent model elements must rise. Simulations, often accompanied by visualization, are preferred to models with first-order physics or simplified rules-of-thumb. Typically, the requirement for greater fidelity has been met at JPL by adopting stand-alone subsystem tools of established credibility, but with the attendant loss of integration with other system and subsystem models/simulations. The drawbacks here are evident as well: stovepiping or a bucket brigade mentality. The danger exists of results being thrown "over the fence" for other team members to somehow make use of, despite being unaligned with their needs.

A Potential Solution – Distributed PTM

The PTM described in this paper diverges from previous JPL PTM efforts by using a distributed approach to connect medium-to-high fidelity constituent models. As before, it aims to leverage existing best-practice tools in which the subsystem experts have confidence. This time, however, it allows these experts to remain involved in the trade process by letting them retain ownership of their own tools. Collaboration is thereby encouraged. Each tool can be kept in its original language/platform, obviating the need for time-consuming translation. Models are inherently accessible to discipline experts for any modifications or improvements. In addition, the ability to feed domain expertise into the networked tool nodes offers better fidelity and a degree of flexibility for trade studies. These "islands" of analytical capability (whether they be completely automated or augmented by humans-in-the-loop) are bridged using a Web Services data exchange approach based on SOAP (Simple Object Access Protocol). The SOAP technology for exchanging data over HTTP is platform-independent, and has emerged as a standard, albeit in several versions, since 2000.

CONSTITUENT MODELS

The following models are examples of those used in the Mars rover PTM:

1. *APGEN* – Time-based operations resource scheduler and scenario engine
2. *ROAMS* – High-fidelity rover simulation run in Monte-Carlo mode for traverse efficiency calculations
3. *Cost Models* – Stochastic and deterministic models using parametric and grassroots methods
4. *Aerospace Corp. Satellite Orbit Analysis Program* – Orbit suite used to calculate view periods between rover locations and candidate Mars relay orbiters
5. *Telecom Link Analysis Tool* – Model to calculate data downlink capability of rover over various conditions
6. *Power Model* – Model to calculate rover power supply as a function of time and location
7. *Mobility & Structure Model* – Rover scaling and reliability tool that models mobility attributes and power expenditure
8. *Launch Vehicle and Trajectory Tools* – Physics-based tools to constrain launch vehicle and trajectory options.

SYSTEMS ENGINEERING CHALLENGES

During the construction of the distributed PTM, a variety of hurdles surfaced. Foremost, from a systems engineering standpoint, the JPL inventory of existing system and subsystem models/tools did not cover all the demands of the specific large Mars rover application. Although useful tools from the available assortment were immediate candidates for certain functions, a lack of others prompted efforts to build needed capability from scratch. Indeed, required models/tools fell into the following three categories:

- Existing models/simulations that could be used without modification or little modification
- Existing models that required significant modification
- Models that needed to be built.

Of course, even existing models with the desired functionality required some nominal effort in the form of Web Services wrapping, the difficulty of which varied by tool. We discovered that some of the existing tools were conceived less as servers than as clients, at best instigating some negotiations on usage protocols and at worst forcing time-consuming overhauls. The server-client issue was not a problem for those models that had to be built from scratch, a consolation prize of sorts for the task of conceiving and developing them ourselves.

Cooperation was sought from the domain expert owners of the tools targeted for inclusion into the distributed PTM. In most cases, their willingness to allow their tools to be wrapped, integrated, and executed remotely never surfaced as an issue. Examples of extreme helpfulness arose, such as the generous reformulation of an existing subsystem model to better serve the Mars rover PTM's specific need. Indeed, one of the key aims of the distributed PTM is to broaden the inclusion of domain experts in the system trade study by utilizing the very same tools they have authored and/or are familiar with. The experience identified some cultural shifts as subtle prerequisites for the success of the distributed PTM experiment, and caused a recognition of the need to effect institutional change with respect to modeling and simulation.

Our approach to maximize use of existing tools brought along all the complications of orchestrating heterogeneous models of widely varying progeny into a cohesive, cooperative, and meaningful fusion. Like the incessant nagging of an overactive conscience, two issues always lurked in the background – integration efficiency and fidelity matching. Since all the stand-alone subsystem tools were conceived in a fairly independent manner and most were never intended to link with other tools, any hope of encountering easy "building block" or "plug-and-play" system integration remained, as anticipated, an excessively romantic notion. Thus, the temptation often beckoned to just replace troublesome existing tools with surrogates we ourselves could build that would fit painlessly into the PTM. Indeed, it was not entirely clear which approach would allow faster implementation, since the friction of shoehorning an already-developed tool into a role with any deviation from its original focus can compete neck-and-neck with the effort required to construct a made-to-order version. But one of the primary goals of the distributed PTM experiment was to avoid any perceived disenfranchisement of the domain experts, so we adhered to the original strategy despite the temptations and only created new models when none existed.

The other lurking issue revolved around matching the fidelity of the distributed models. Since the existing tools embraced different levels of fidelity, they

required different levels of design information and/or greater or lesser comprehensiveness in their input parameters. This obviously reverberated across the PTM. In practice, the solutions to fidelity issues typically resulted in seeking the path of least resistance while maintaining goal integrity ("Better is the enemy of good enough.").

USE OF A FUNCTIONING TOOL OF TOOLS

At the heart of the constellation of distributed models resides the chief controlling element, coded in Excel and Visual Basic, and referred to as the Integrated Summary (IS). In it, the system engineer/analyst interested in performing a trade can select among available inputs to initiate the analysis of a new alternative. It is important to realize that the analyst does not have explicit access to every causal parameter that affects the output; rather, detailed subsystem-specific controls are kept within the domain of the distributed models while the IS holds only the top-level design levers. Apart from the inputs, the IS also captures manually and automatically-entered metadata like rationale, assumptions, and configuration management information to enable effective future navigation through an expanding library of results. These results, the targeted outputs of candidate system alternatives calculated during a successful run, appear in the IS neatly correlated with their associated inputs and metadata. Figure 3 shows a simplified representation of a tradeoff comparison with multiple alternatives as displayed in the IS columnar format. In addition, the enumerated tradespace can be displayed graphically, with different MOEs plotted against LCC.

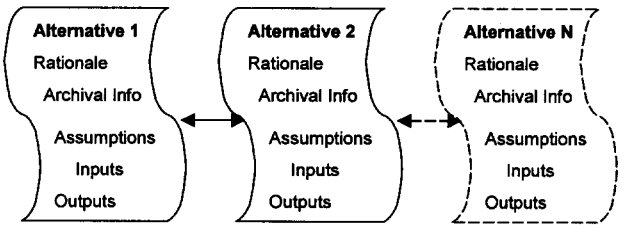

Figure 3 — Integrated Summary Layout

To initiate a run, the system engineer/analyst selects the top-level inputs, adds archival notes, and inaugurates the computational cascade. The IS then sends requests to the various remote tools distributed across JPL and waits for their response. The dissimilar computational speed and variegated complexity levels of the domain expert models cause different arrival times for the results.

The PTM can be run in a stand-alone mode with a single system operator or in a concurrent mode with project team members. In the stand-alone mode, the detailed subsystem-specific input parameters residing in the distributed models are treated as givens, whereas in the concurrent mode, the domain experts can, in real-time, adjust them as needed for each alternative. The stand-alone mode offers the capability to perform a greater quantity of trades within a given time without obligating a team for the session, but the concurrent mode gives more nuanced analyses and potential fidelity benefits using humans in the loop.

A particular challenge for a distributed system of models is managing and controlling the configuration. Constituent tools can be upgraded with new or revised functionality, and assumptions or parameters within them can easily change. The PTM handles this problem on two levels: with version control on individual tools and a combinatorics approach between them. This allows PTM trade results to be traceable to the exact inputs used, a non-trivial undertaking when model control is partially decentralized. Agreements with the team over standards of participation greatly help in this regard.

ISSUES RAISED

While building our distributed Mars rover PTM, issues inevitably arose regarding the architecture that should be imposed on our collection of tools/models. One such issue concerned the proper role of the IS in relation to the distributed models–that is, how much control should the IS have over the ensemble? Two separate identified positions provide a convenient framework to explore the matter; we call these two positions the "strong" vs. "weak" PTM.

A "strong" PTM, shown in Figure 4, has two distinctive aspects. First, the distributed models are arranged as spokes around an IS hub. Each model is queried independently by the IS and returns its results reciprocally. Second, all the design variable inputs for the tradeoff study originate exclusively within the IS. Therefore, the amount of information sent to the models is comparatively large (shown by the thick arrows) since every parameter that affects the distributed computations is controlled centrally.

The other extreme is the "weak" PTM, shown in Figure 5. In contrast to the "strong" PTM, the "weak" PTM's models are allowed to interact with each other, passing information in a network style. Some models may never exchange data directly with

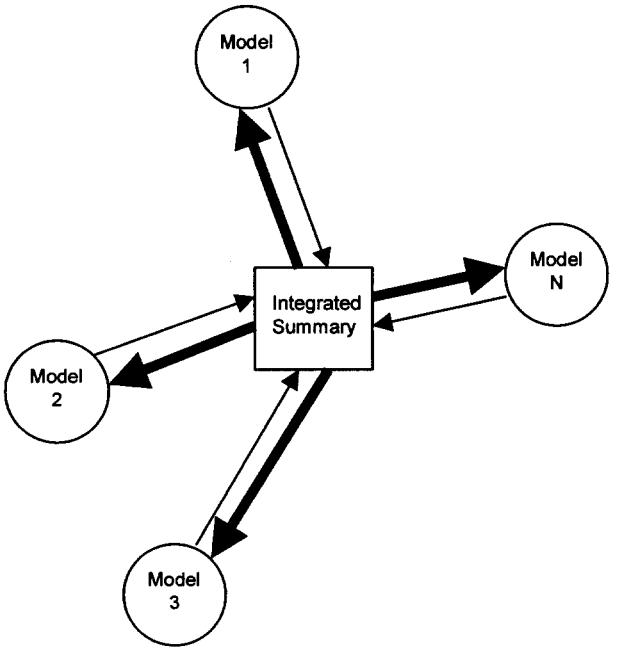

Figure 4 — The "Strong" PTM Architecture

the IS, but be activated as subroutines by models upstream in the data flow. In addition, some of the design variable information for the tradeoff study is kept within the distributed models, leaving the IS in control of only a portion (although typically the top-level and most important) of the inputs. The amount of information sent from the IS to the models is thus comparatively small.

Each approach retains certain advantages over the other. The "strong" PTM concentrates the system engineering task within the IS, where calculation threads using individual model results can be woven under the control of the PTM builder, avoiding diffusion of that critical responsibility. The bilateral relationship between each model and the IS facilitates the maintenance of an easily communicated, rigorous input-output interface, offering a straightforward framework for block-replacement model upgrades. This, combined with the inherent traceability of the data flow (where the origin of parameter inputs is confined to one source) ameliorates the configuration management problem. In addition, with unhindered access to all system variables, classic methods can be applied for mathematical optimization.

The "weak" PTM relieves the IS from the burden of bookkeeping all the system parameters, allowing a simpler IS with leaner interfaces. The reduced complexity arises from a smaller data set to handle, some delegation of inter-module calculation threads, and the avoidance of store-and-forward routing issues. The system engineering task is not concentrated exclusively in the IS, but rather in the arrangement of models. The "weak" PTM offers a more realistic opportunity for humans in the loop at each node to review and modify parameters before sending them to the next destination. The distribution of control also bolsters a perception of inclusivity amongst team members.

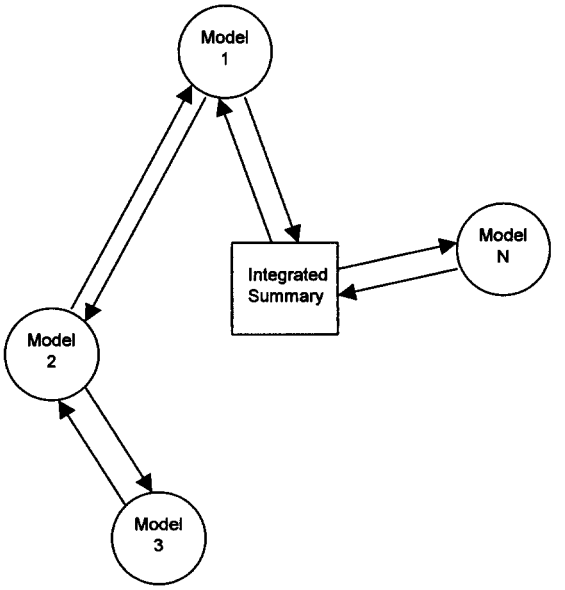

Figure 5 — The "Weak" PTM Architecture

The "weak" PTM concept enables some tantalizing theoretical advantages in the future. If domain experts conceive and implement later versions of their tools as servers, with the functional computations continuously available over the web to potential users, the time to gather subsystem outputs could diminish dramatically. From a library of web-enabled smart tools, elements could be rearranged to solve new problems as long as all input requirements are met. The requisite information could be supplied from the best available source, whether straightforward user inputs or the latest version of another model. Interconnectivity would not be problematic when all such tools adhere to Web Service protocols. The approach necessitates, of course, that model assumptions are transparent to all participants. Coding models in such a manner to maximize interface flexibility would result in greater upfront costs, but cost savings should result when these models are reused on another project.

The Mars rover PTM was implemented as a "weak" PTM. This was a natural outgrowth of the approach

to maximize use of existing tools with the least amount of modification. Indeed, most existing stand-alone domain models had been conceived under the assumption that engineer analysts would handle inputs and commands rather than having those passed automatically via electronic means; in short, these models were not designed to function well in the role of slavish nodes. Despite the additional complexity of harder configuration management, and a more demanding system engineering task, the "weak" PTM was more aligned with the overriding principle of fostering enhanced team inclusion. This reason, combined with the desire to explore the promising future benefits of Web Service-enabled smart tools, led us along the chosen experimental direction.

CONCLUSIONS

Our experience in developing a distributed PTM for a large Mars rover mission suggests that the approach is a viable method for conducting other complex space mission tradeoff studies. While it is premature to champion distributed PTMs as preferable to single-platform PTMs, the ongoing results of the experiment show considerable promise. At this point, some clear lessons have emerged along both technical and organizational lines.

The technological underpinnings of the distributed PTM concept are demonstrably sound. We found Web Services up to task, although extra vigilance was required to check the connection status between distributed tools. Recently emerged software packages now available commercially hold promise to expedite the process of wrapping existing tools. Expanding the extent of connectivity for cross-NASA Center collaboration appears technically feasible, (assuming resolution of firewall issues), and would be a logical progression of the distributed PTM concept.

From the beginning, it was clear that a new approach to trade studies might require shifts in established organizational practices. We maintained a focus on easing this transition by adhering to the "weak" PTM structure with its collaborative appeal, even when tempted to concentrate more control for greater implementation efficiency and simpler oversight. A clear recognition emerged of the need to effect institutional change with respect to modeling and simulation. Combined with the enabling technology, this maturing idea of tool and process cooperation will lead to more timely ways of providing rigorous trade study results for smart decisions.

ACKNOWLEDGEMENTS

The authors would like to thank the members of the PTM team at JPL, Mars rover domain experts who offered their tools for the experiment, and the supportive members of management who encouraged new approaches to systems engineering. The financial support of the National Reconnaissance Office and the Mars Science Laboratory Project Office is also acknowledged.

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

REFERENCES

[1] Processes for Engineering a System, EIA-632, Electronic Industries Association, January 1999.

[2] Systems Engineering Fundamentals, Supplementary Text prepared by the Defense Acquisition University Press, Fort Belvoir, VA 22060-5565, December 2000.

[3] Sage, A., and Armstrong, J., *Introduction to Systems Engineering*, John Wiley & Sons, Inc. 2000.