Integrated Acquisition-Logistics Synthetic Environments for Total Value Assessment

Reuse and Interoperability of Virtual Products Key to Payoff on a National Scale

GARY JONES • HENSON GRAVES • MARK GERSH

Total Value Assessment for the acquisition, delivery, provisioning, and sustainment of warfighting forces requires a greater understanding of how these processes develop, interrelate, and evolve in real-world situations. These processes are complex and tend to depend upon an immense amount of data. In many cases, small changes in the environment produce large discontinuous changes in the way the processes work. For example, estimating the amount of supplies needed for Desert Storm, actually shipping them, and returning the unused items afterward stressed our ability to predict and control the supply process, and resulted in quite a few undesirable ripple effects.

Vision

Creating models for acquisition-logistics processes and simulating their execution within a synthetic environment provides the best tool available for assessing the total value of products and their associated processes. Within a synthetic environment, we can instrument and monitor an unfolding process and its constituent product(s) to gather data for later analysis; or in real time, interactively ask "what-if" questions by making adjustments to the product(s) and process(es) to better understand resultant behavior.

> **Creating models for acquisition-logistics processes and simulating their execution within a synthetic environment provides the best tool available for assessing the total value of products and their associated processes.**

"Estimating the amount of supplies needed for Desert Storm, actually shipping them, and returning the unused items afterward stressed our ability to predict and control the supply process, and resulted in quite a few undesirable ripple effects." Pictured is a warehouse at the Port of Jeddah, Saudi Arabia, containing supplies for Operation Desert Storm. The supplies were loaded onto Military Sealift Command ships for transport to the theater of operations.

A major DoD Modeling and Simulation (M&S) objective is to perform virtual warfare engagements using simulated and actual weapon systems. This vision and its objectives can be broadened to include the acquisition and logistics processes of simulated and actual systems. For example, a logistics planning exercise using weather and climate data may link to actual operational supply vessels and into

Jones is the Program Manager for the Simulation Based Design Program of the Defense Advanced Research Projects Agency (DARPA) in Arlington, Va. Graves is the Deputy Program Manager for Technical Operations, Simulation Based Design Program, Advanced Technology Center, Lockheed Martin Missiles and Space, Palo Alto, Calif. Gersh is the Program Manager, Simulation Based Design, Lockheed Martin Missiles and Space, Advanced Technology Center, Palo Alto, Calif.

Photo by PH2 Milton R. Savage, USNR

commercial transport systems so as to assess the trades of augmenting DoD systems with commercial delivery systems. Imagine if this same exercise included connections to models and simulations of the transported products so the issues of retrofit and manufacturing could also be addressed, giving logisticians an even more complete or total value assessment of all options.

Achieving the Vision

Achieving the vision of integrated acquisition and logistics synthetic environments still presents a number of technology challenges. Before virtual engagements can be of most value to life-cycle analysis, they will require high-fidelity system models. During the course of analysis, we need an ability to refine components of high-level aggregate models into detailed high-fidelity models to better explore specific aspects of a life-cycle problem. M&S is already used for planning and warfare analysis at different levels of abstraction (campaign, engagement, and system interoperation); however, current M&S systems have little ability to integrate multiple-fidelity models into a simulation exercise.

No single organization will be able to build and maintain the collection of virtual prototypes needed for these exercises. Since prototypes will be built and used by many different organizations, achieving interoperability requires the use of at least de facto standards and perhaps an organization to promulgate those standards. Currently, some standards are beginning to emerge for representing virtual prototypes. For example, modelers explicitly designed Virtual Reality Modeling Language (VRML) to produce virtual prototypes that can be placed in synthetic worlds and interact with other objects in these worlds. This emerging technology needs to be integrated and more exploited within the acquisition-logistics community.

Constructing and performing assessment exercises in a synthetic environment requires a distributed modeling and simulation framework in which a user can discover and configure virtual prototypes, then launch exercises without human involvement at any of the distributed sites that contain prototypes. Commercial technology and standards that address tool-to-tool communication (e.g., Common Object Request Broker Architecture [CORBA], Internet protocols) are available and can be exploited for the assembly of distributed synthetic environments.

SBD's Influence/Accomplishments

The Defense Advanced Research Projects Agency Simulation Based Design (SBD) program is developing a prototype distributive, collaborative software system that addresses some of the functions required for fielding the types of integrated acquisition-logistics synthetic environments that support the development, analysis, and interoperation of virtual prototypes.

Previously, DARPA's SBD program validated the feasibility of establishing distributed synthetic collaboration environments between multiple heterogeneous organizations. To meet new threats, these collaborative synthetic environments used *engineering* analysis to better evaluate operational warfighting performance and used operational analysis to reengineer weapon systems.

Engineers also used SBD to develop conceptual design models and detailed engineering models for ships. Of sufficient structural detail that modelers can use them for parametric design optimization, the conceptual models can be placed in high-fidelity operating environments. The detailed models have been used to generate shop floor manufacturing instructions and to provide immersive maintenance training.

During the past year, the SBD program performed a validation experiment, called the Advanced Surface Combatant (ASC), that culminated in a February 1997 demonstration of SBD maturity. This experiment specifically focused on the survivability analysis and redesign of a surface combatant to meet a new threat. It also provided the opportunity to include detailed physics-based models in the warfighting analysis phase and use of multidisciplinary optimization techniques to provide parametric design information to the redesign process.

The ASC Experiment resulted in an SBD system configuration that —

- integrated multiple companies and government agencies into an Integrated Product Team (IPT);
- organized the IPT as a hierarchical collection of federations;
- operated over a combination of Local Area Network, Internet, and DARPA gigabit testbed (ATDNet) network resources;
- integrated approximately 30 software components into the system;
- integrated two legacy databases and ingested the indicative design of the Navy SC-21;
- provided interface code that wrapped legacy simulations making them compliant with the DMSO High Level Architecture;

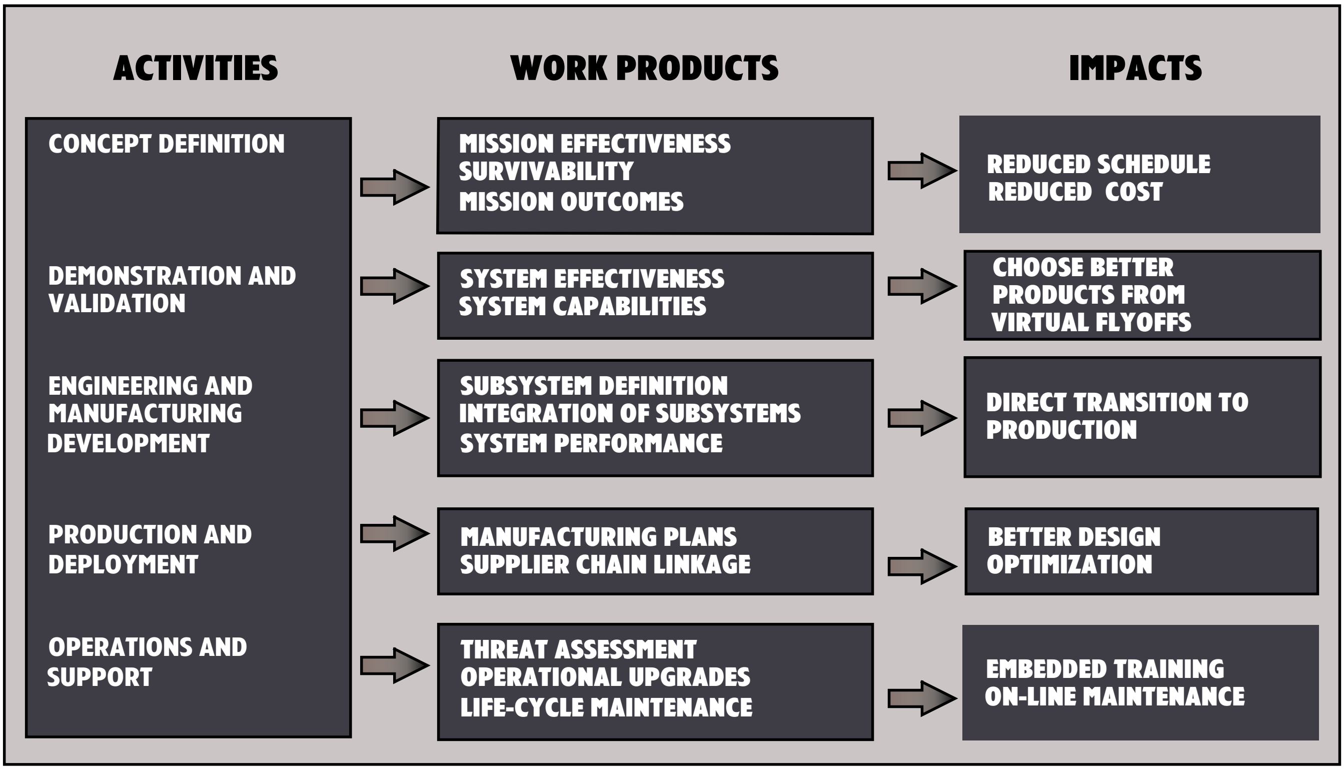

Figure 1. **Product Life-Cycle Activities, Work Products, and Impacts**

- demonstrated the use of SBD in multiple life-cycle activities (from requirements to training);
- demonstrated the use of multidisciplinary analysis and optimization; and
- incorporated cost as an independent variable in the design trades.

The ASC experiment represented a significant achievement in maturing SBD technology to the stage that it can now be deployed for experimental use by contractors and the government for conceptual system design, development, and evaluation.

The SBD Product

The SBD system is a collaborative, multidisciplinary environment for developing and using virtual/real prototypes. Engineers configure an SBD system for a specific application by linking copies of a common set of software, called the Core Processing System, together with application-specific software tools. SBD allows engineers to develop, analyze, and operate virtual prototypes as they would actual prototypes, but without the cost and complexity associated with real hardware and materials. A virtual prototype is a computer software module that models the structure and behavior of the actual product under development. The process of producing an actual product proceeds as a series of virtual prototypes that defines the product and/or generates manufacturing instructions for the product. For the virtual prototyping process to yield actual quality products, the same disciplines must be applied in the virtual prototyping process as are applied in conventional product development processes.

Numerous integrated development environments exist, tailored to a selected computer-aided design (CAD) tool (e.g., the Boeing 777 CATIA™-based environment), and many organizations have now integrated modeling, simulation, visualization, and analysis tools for product development. However, engineers craft these concurrent engineering systems for specific applications, which significantly limits their reusability, even between different projects in the same organization. Developing the second system becomes as expensive as developing the first. Further, there are no standards to allow these different systems to interoperate.

The SBD process employs a much more open approach that produces a variety of design, engineering, and evaluation results. Figure 1 illustrates the product development activities, SBD work products, and their impact on the product life cycle. This process delivers better quality products at a reduced cost, risk, and schedule when compared to the current, more conventional concurrent engineering approaches. The virtual prototyping activities can be conducted in a distributed collaborative software environment, which allows more concurrence in the development tasks, thus reducing schedule slippage.

Using virtual prototypes for engineering analysis and operational validation also allows for investigation of larger solution spaces. Changes can be made much later in the virtual product development life cycle without incur-

Figure 2. **SBD Integrates Multidisciplinary Life-Cycle Activities**

ring the cost magnitude that changes further downstream make in conventional development processes.

Figure 2 illustrates the life-cycle activities for a notional ASC Navy program, integrated and supported by a collaborative SBD system configured as an IPT. The IPT involved multiple government and contractor organizations with participants for program management, design, engineering analysis, operational test and evaluation, and deployment and training.

The ASC SBD system linked multiple copies of the common software components together to support the IPT. Engineers configured each participant's software to reflect one of the following four roles in the life-cycle development process:

- Program Management Office
- Hull Mechanical and Electrical Design
- Combat System Design
- Survivability Analysis

While some of the ASC software was ship-specific, much was domain-independent and could be used for other application areas. The experience of integrating such an SBD system translates readily to other domains.

Using SBD

With SBD, a user can define, modify, visualize, and manipulate virtual products. The SBD system coordinates the management between multiple user activities by using virtual prototypes that are composed as assemblies of subsystems and parts. Engineers define the actual construction of parts in terms of material, structure, and behavior attributes. By combining legacy models in various ways and by producing data that can be used by a variety of legacy analysis tools, they construct virtual products. The values then, of these attributes may be computed by external tools or incorporated from legacy databases.

Users access SBD through a standard web browser. Figure 3 shows a satellite prototype as viewed from an early SBD User interface prototype. Since a key feature of the user interface is its use of standard web browsers, it can easily use standard plug-in tools such as VRML viewers to display a wide spectrum of standard data types.

This particular user interface prototype lacks the elements for controlling analysis and design tools, but it does show how engineers can easily access information about the design elements. In this example, the window on the right shows a component hierarchy of the satellite and allows the user to access components like bus structure, power, propulsion, attitude determination and control system, thermal, and payload modules. Each of these components has its own decomposition, and the subcomponents are interconnected in various ways. Connections are maintained as part of the product definition.

The window on the left displays the satellite as viewed within a 3D visualization and interaction environment. This satellite prototype responds to a set of commands that can be used to deploy its solar panels and actuate mechanical devices on the satellite. Operating the satellite within this kind

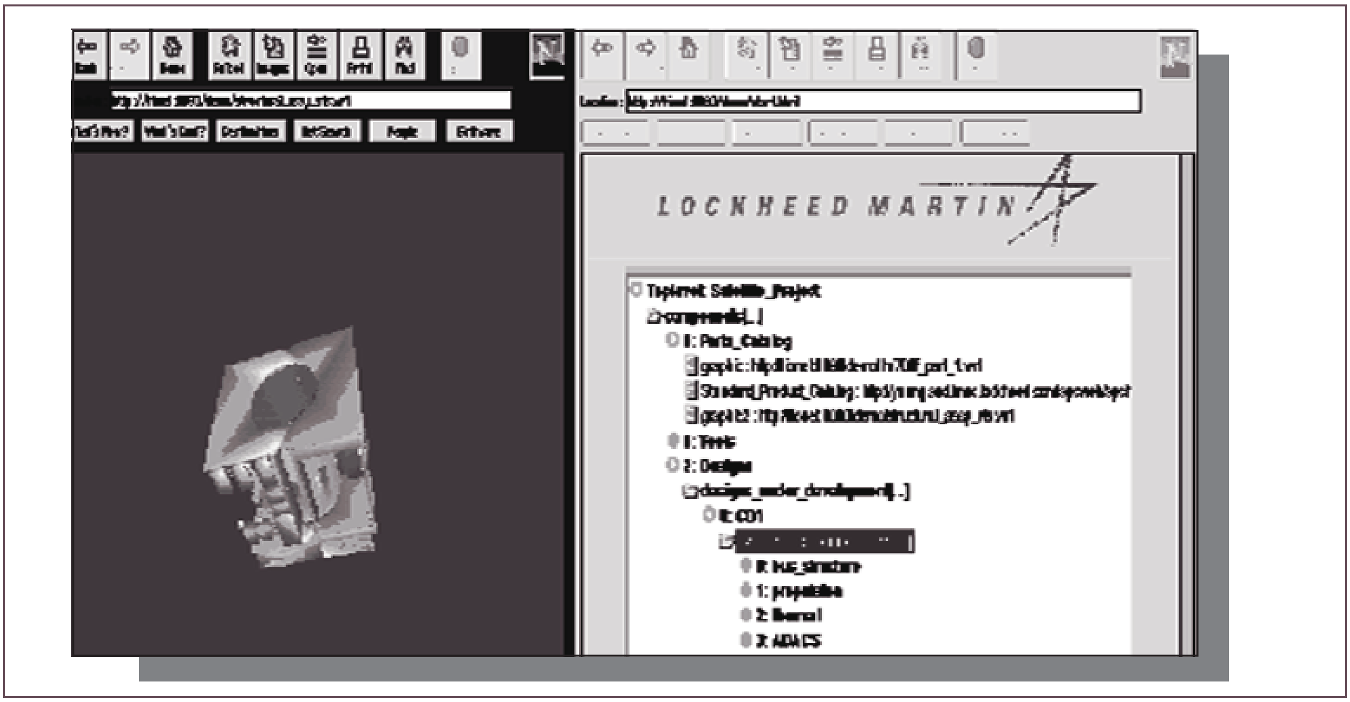

Figure 3. **Satellite Prototype Viewed from the SBD Guided User Interface (GUI)**

of environment can be used to evaluate a number of design properties such as sensor field of view, inter-satellite communications, ground station communications, mechanical interference properties of deployment and actuation devices, and advanced technology insertion. In this example, engineers used VRML to produce the visual appearance of the satellite, with the VRML being computed from the virtual product representation.

The specifics upon which modelers define virtual prototypes differ, depending on the level of fidelity needed and the data requirements of the tools used to evaluate the prototype. Easy to modify and clone, engineers can later reuse virtual prototypes as parts of other prototypes. New attributes (e.g., center of gravity) can also be added to a prototype when needed for a particular application.

Further, engineers can quickly generate an initial conceptual design to validate feasibility and provide a basis for cost estimation by reusing data from previous systems and by importing data from external tools. The virtual products can then be analyzed with existing or legacy analysis tools, and can be operated in virtual environments combining real and simulated products.

With SBD, engineers can also capture design processes, such as the steps in designing a power subsystem for the satellite, as mega-programs – or programs of programs – that are manipulated and operated exactly like the virtual prototypes.

Development proceeds within SBD by establishing product constraints and requirements and by constructing increasingly detailed virtual prototypes of the product. The virtual prototypes (software models) can be viewed, interacted with, analyzed, and operated like real prototypes. As engineers make design changes, analysis and evaluation of the prototypes takes place within synthetic physics-based environments. SBD not only manages these design artifacts, with built-in configuration management tools, but also allows engineers to incorporate components of different levels of fidelity within a virtual prototype.

Collaborating With SBD

Complex product development typically involves multiple heterogeneous organizations. The inter-connectivity needed for product development requires support for defining, managing, and enforcing development processes and the resulting workflow. In a large-scale product development enterprise, each team has its own data, product, and process models.

Since engineers configure a collaborative SBD system as a collection of copies of the common software, it works to provide seamless access to all public resources in the entire SBD system. Each user interface provides access to the rest of the SBD system, as mediated by the Core Processing System components.

SBD allows users to maintain their product data in one or more databases – legacy or new, flat file, relational, or object oriented – which can be either centralized or distributed. Each Core Processing System maintains an object model that is accessed from its user interface, for visualization and interaction.

A Project's View of SBD

SBD uses object models to represent all product and process information. Within an enterprise's organizational hierarchy, object models, collectively called the Smart Product Model

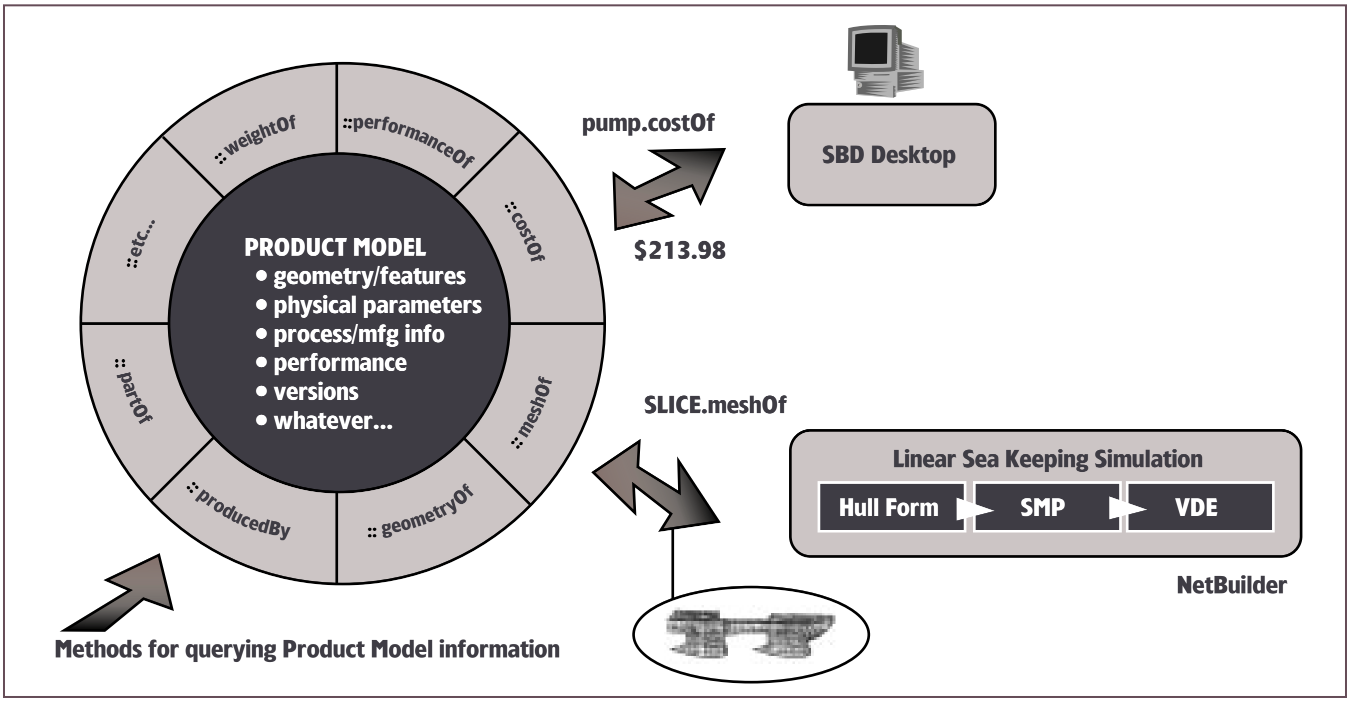

Figure 4. **A Logical View of the SPM**

(SPM), define and manage the development process, the external software and databases that are part of a specific SBD configuration, as well as the product models that engineers develop as part of the life-cycle process.

The SPM may be viewed as a collection of concentric circles with data at the center; "smart" methods that directly operate on the data in the innermost ring; software components that provide value-added services such as 3D viewing and interaction with product model data as the next ring; and finally, external programs that interface to the core data as the outer ring.

Figure 4 illustrates this view of the architecture for a notional ship design project.

This architecture is a natural extension of the single CAD model approach used on large programs such as Boeing 777. Integrating behavior, management, and analysis data into a single virtual enterprise-wide distributed data model ensures that all members of the team always have access to all information relevant to their design, and that the impact of design or management changes – such as schedules or budgets – can be immediately assessed by all team members.

The object models are "smart" because they have methods that are used to perform analysis and other development activities. Methods are the means to manipulate or analyze data such as meshing of CAD data for structural analysis, aerodynamics for aircraft maneuverability, or a seakeeping model for ship motions. As such, they can be aggregated to form views into the object model that are specific to a given discipline or user group. As an example, methods may be used to calculate the weight of an object as the sum of the weights of its components; or methods can be used to expose a data view relevant to the structural design engineer.

Conclusion

SBD is the first step toward fielding integrated acquisition-logistics synthetic environments. By harnessing advancements in M&S, High Performance Computing and Communications, and Multimedia technologies, SBD provides a virtual collaborative environment for geographically distributed IPTs to design complex systems and provide support throughout the product's life cycle.

Today, M&S is becoming increasingly important in acquiring systems for the government, but the potential cost reductions offered by correctly using M&S in the design process still dwarf deployed reality. Why hasn't the DoD acquisition community yet realized these substantial cost reductions? The answer is contained in the following three problem areas: tools don't interoperate, people are in the loop even when no decision-making requirements exist, and no standards for digital product and process models exist. SBD offers a solution to these problems by leveraging emerging standards and commercial forces for interoperability, by fielding a collaborative software environment infrastructure, and by creating de facto standards for product and process models.

The SBD program is unique in developing a virtual prototyping architecture for configuring reusable and interconnective SBD systems with standards-based interfaces. The ability to reuse and interoperate virtual products across multiple organizations and vendors is where SBD will pay off on a national scale.