CHAPTER 9

Simulation-Based Acquisition

STEPHEN R. OLSON and ANDREW P. SAGE

9.1 INTRODUCTION

The decade of the 1990s was a tumultuous time for the U.S. Department of Defense (DoD). It was a time that not only gave birth to a new defense posture for the country but also demanded change in the methods used to develop and produce large and complex systems for national defense. Brief comments on these two issues are of importance to simulation-based acquisition (SBA).

9.1.1 Background: U.S. Military Posture (1980 vs. 2001)

The 40-year Cold War (1950–1990) was a period of significant world tension and confrontation between the communist countries led by the Soviet Union and democracies generally led by the United States. In response to the posture of the USSR, the United States invested heavily in the development of increasingly complex aircraft, ships, tanks, and global surveillance systems. The Cold War arms race culminated in what is sometimes termed the "Reagan Build-up" of the early 1980s, the subsequent collapse of communism in Europe, and the disintegration of the Soviet Union by 1990. In practically a few months, the Cold War abruptly ended. There was much to be thankful for and the public sentiment in the United States quickly turned in favor of decreased spending on national defense. While spending commitments and long-term procurement plans prohibited an immediate collapse of expenditures, by 1993 spending on the development and procurement of military equipment had decreased by more than 50 percent from the peak in 1985 of about $175 billion per year expenditures to about $80 billion. It has remained essentially constant from 1993 through 2000. At the same time, the disintegration of Soviet influence in Europe and the Middle East gave rise to new sources of global instability and threats to regional peace. While different, these new threats confounded the complexity associated with the bilateral confrontation pattern of the Cold War. The U.S. defense establishment was confronted by the need to reconsider almost every aspect of how it planned and

Handbook of Human Systems Integration, Edited by Harold R. Booher.
ISBN 0-471-02053-2

equipped for national defense. These changes in the world environment and the repercussions on national defense have come to be called the *revolution in military affairs* (RMA).

While rethinking the national defense strategy occupied many defense planners during this time, there was another equally complex problem confronting the DoD. During the Cold War the United States was a leader in developing and applying electronic technology to address defense needs. Initiatives funded by the DoD covered the spectrum from the manufacture of reliable chips and components to the design of "supercomputers" and the architecture of complex software and communications systems. As costs associated with these maturing technologies fell dramatically, their commercial potential blossomed, and the 1980s witnessed an extraordinary commercial growth in digital and communications technologies, a pattern that has continued into the twenty-first century. Commercial investment in these new-age information technologies rapidly outpaced that of the DoD. At first glance, this might seem a piece of good fortune, in that it became possible in principle for major cost savings to result from leveraging these commercial investments through use of commercial off-the-shelf (COTS) products. However, modern commercial business practices in contemporary high-technology industries generally bear little resemblance to the DoD business practices of the Cold War. Today, the great preponderance of commercial electronics and software and the associated information technology products have a life of only a few years. Furthermore, the information technology industry has developed a business model that makes it often insensitive to penalties associated with product defects and errors through continual release of upgrades and reengineered products, a situation not acceptable for military weapons systems. As the commercial market for digital components has expanded, manufacturers may have little interest in the relatively low production quantities required for unique military systems. Today, even a relatively new military system has components that are obsolete, with a very limited supply of compatible spare parts available in the market. Many defense systems, such as ships, aircraft, and tanks, experience operational lives of 20 to 50 years, and the effort and expense in maintaining the technological currency of these systems have proven to be a challenge. This problem has been compounded by the significant reduction in defense spending noted earlier. By 1993, the United States had practically ceased the acquisition of major new defense equipment, investing the bulk of DoD funds in the modernization and maintenance of existing equipment. To some degree, this strategy was acceptable, particularly given the large quantities of new equipment still in the production and delivery pipeline from the Reagan years. The result, however, by the year 2001 is that we had a rapidly aging fleet of ships, planes, and weapons of all types.

All of these issues have served to remind the DoD that the total cost of system ownership is dominated by operating, maintenance, and support costs that occur years after product acquisition (Buede, 2000). However, there is great difficulty in accurately projecting these future costs and in minimizing their impact during the product development phase. Furthermore, the need to change a product in an evolutionary manner after it is fielded is increasing, because maintenance of technological currency is essential to maintaining combat effectiveness, while the rate of technology "innovation" continues to escalate. These problems emphasize the need to anticipate the retrofit of new technology into fielded systems so that product upgrading can be planned and accomplished in a cost-effective and timely manner. The obvious question with all of these issues and with little or no prospect of funding ever returning to the levels of 1985 yet with demands for military presence and peacekeeping across the globe is, "How can the United States sustain its defense capability in a trustworthy manner?"

All of these factors have caused defense planners to realize that the DoD can no longer continue to develop and acquire systems as it has in the past. Just as the change in the world balance of power and U.S. military posture has generated the need for a RMA, this change in the business landscape caused many to declare that the United States also needed a revolution in business affairs (RBA), often referred to more modestly as *acquisition reform*.

9.1.2 Background: Motivation for Revised Acquisition Practices

A large number of problems have been encountered with "grand design" or waterfall life-cycle efforts traditionally used to engineer a system. Thus, there have been a number of efforts to extend developmental approaches beyond the classic waterfall approach (Sage, 1992, 1995; Sage and Rouse, 1999). Today, the classic waterfall approach is suggested only in those rare cases where user and system-level requirements are crystal clear and unlikely to change and where necessary funding for all life-cycle phases associated with the grand design is essentially guaranteed. This is rarely the case for major systems, especially those that are software intensive. Changing user needs and technology virtually guarantees that major systems cannot be developed using the grand design approach.

Two leading alternative approaches to the engineering of systems are termed incremental and evolutionary. Incremental development has as a plan to deliver the system in preplanned phases or increments, in which each delivered module is functionally useful. In such an approach, the overall system capability improves with the addition of successive modules. In such an approach, the desired system capability is planned to change from the beginning as the result of "build N" being augmented and enhanced through the phased increment of "build $N+1$." This approach enables a well-functioning implementation to be delivered and fielded within a relatively short time and augmented through additional builds. This approach also allows time for system users to thoroughly implement and evaluate an initial system with limited functionality compared to the ultimately desired system. Generally, the notion of preplanning of future builds is strong in incremental development. As experience with the system at "build N" is gained, requirements changes for module $N+1$ may be more easily incorporated into this and subsequent builds.

Evolutionary life-cycle development is similar in approach to its incremental complement; however, future changes are not necessarily preplanned. In this approach, we recognize that we are unable to initially predict and set forth engineering plans for the exact nature of these changes. The system is engineered at "build $N+1$" through reengineering the system that existed at "build N". In this approach, a new functional system is delivered at each build, rather than obtaining "build $N+1$" from "build N" by adding a new module. The enhancements to be made to obtain a future system are not determined in advance, as in the case of incremental builds. Evolutionary development approaches can be very effective in cases where user requirements are expected to shift dramatically over time and where emerging and innovative technologies allow for major future improvements. It is especially useful for the engineering of unprecedented systems that involve substantial risk and allows potentially enhanced risk management. Evolutionary development may help program managers adjust to changing requirements and funding priority shifts over time since new functionality introductions can be advanced or delayed in time in order to accommodate user requirements and funding changes. Open, flexible, and adaptable system architecture is central to the notion of evolutionary development. As a follow-on to this, it appears that evolutionary development of a system architecture has the potential to greatly decrease the risk and costs of excessive rework of a system of systems or a federation of systems after it has been initially engineered. Figure 9.1 indicates the general nature of the evolutionary life cycle. This can be represented as a continuing waterfall, with feedback across the life cycle or as a spiral. Much of what has come to be known as evolutionary acquisition is based upon an equivalent spiral life cycle (Boehm and Hansen, 2001).

The DoD has not been unmindful of these needs and the need for evolutionary life cycles. Incremental life cycles were recognized a decade ago and made a part of the DoD 498 standard, which is no longer operational due to the decision to use commercial standards whenever feasible. Acquisition reform is a major effort now and has been for much of the past decade. In the effort to reduce acquisition response time, the rewrite of the DoD 5000 series regulations (DoD, 2000a) calls for evolutionary acquisition to be the preferred method for future defense acquisition programs. It also calls for SBA to support this. Unfortunately, there is often considerable confusion over the meaning of these terms and life-cycle development methods that should be used in the pursuit of various evolutionary acquisition and simulation-based acquisition approaches. Some of this mystification is evident in the use of expressions such as evolutionary development, spiral development, spiral acquisition, evolutionary spiral development, and a host of other expressions where the meanings are not well understood and accepted across those using the terms.

There are a number of follow-on evolutionary acquisition efforts. Evolutionary acquisition strategies define, develop, and deploy an initial, militarily useful capability and a plan for subsequent definition, development, test, and production/deployment of increments beyond the initial capability over time. The scope, performance capabilities, and timing of subsequent increments shall be based on continuous communications among the requirements, acquisition, intelligence, logistics, and budget communities.

An excellent overview of evolutionary acquisition may be found in a Defense Systems Management College (DSMC, 1998) report. There it is indicated that evolutionary

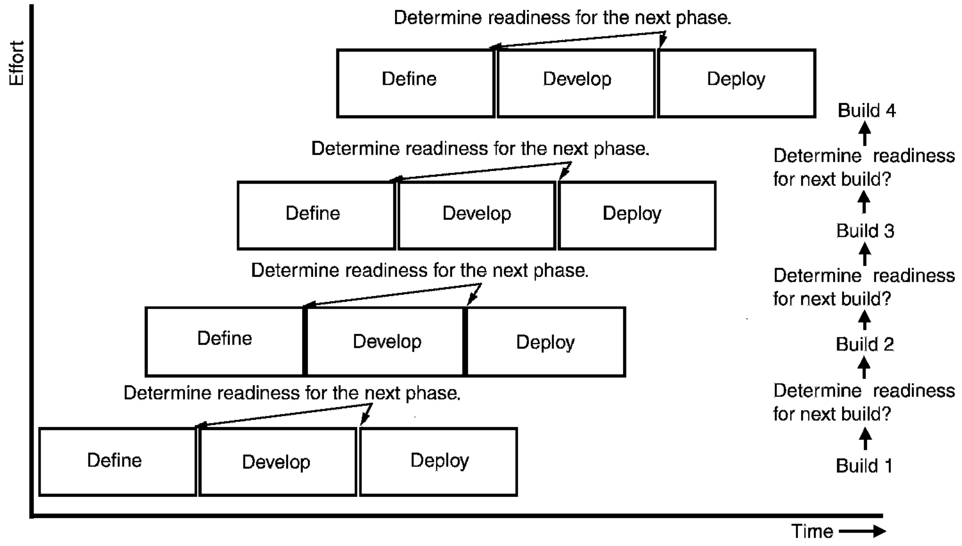

Figure 9.1 Iterative life cycles in evolutionary acquisition.

acquisition is a strategy for use when it is anticipated that achieving the desired overall capability will require the system to evolve during development, manufacturing, or deployment. This appropriate definition provides a suitable linkage between the concepts of evolutionary acquisition and complex adaptive systems through use of the term *emergence*.

It is important also to recognize the complex adaptive nature of today's technologies and organizations. Interestingly, most studies of complex systems often run completely counter to the trend toward increasing fragmentation and specialization in most disciplines. It is not at all a large number of parts in a system that makes the system complex; it is the way that the parts interact. A product may consist of abundant parts, but if these parts interact only in a known, designed, and structured fashion; the system is not complex, although it may be big. Complexity exists when the interconnected parts of a system interact in unanticipated ways. One of the defining characteristics of complex systems is the property known as emergence. Here, the behavior of the overall system is different from the aggregate behavior of the parts, and knowledge of the behavior of the parts will not allow us to predict the behavior of the whole system. The emergence property is a form of control. It allows distributed agents to organize together to determine consequential higher order system behavior. In systems that are "complex," structure and control emanate or grow from the bottom up. Thus, the reductionist scientific approach generally does not work with complex systems. Virtually all organizational behavior in such systems is comprised of agents adapting to their environments and, in the process of so doing, affecting the environments of all other agents. In some situations, when systems are driven sufficiently far from equilibrium, *bifurcations* occur and chaotic behavior may result.

Clearly, there are many considerations involved in efforts such as these. The prototypical steps in building an experimental and exploratory model of a complex adaptive system might be described as follows:

1. Simplify the problem as much as possible, being sure to retain the essential features of the situation.
2. Identify a potentially appropriate model of the situation that represents agents that follow simple rules with specified interactions and randomizing elements.
3. Construct a simulation based on this model.
4. Run the simulation many times with appropriately different random variables, collect the data, and compute statistics from the different runs.
5. Identify how simple behavioral rules result in observed behavior.
6. Study the responses obtained by sensitivity studies and appropriate parameter changes to determine critical parameters, sources of behavior, and effects of different parameters on system responses.

There is a major role for modeling and simulation in the several activities suggested in this list. This creates a strong linkage between evolutionary acquisition and SBA as a way to study evolutionary acquisition and other acquisition phenomena.

9.2 OBJECTIVES FOR SBA

Simulation-based acquisition involves much increased use of computer-based models and simulations within system engineering and product acquisition life cycles. It is an acquisition process enabled by modeling and simulation technology integrated across acquisition phases and programs. It has the objectives of reducing the time, resources, and risk associated with the acquisition process while improving the trustworthiness and supportability of deployed systems.

Before proceeding further, it is useful for us to introduce some definitions appropriate to the "simulation world." There are a significant number of terms and associated definitions used in SBA. A relatively thorough list may be found in *Acquisition Modeling and Simulation Comprehensive Core Body of Knowledge* (Acquisition Functional Working Group, 1999), which also contains extensive references to the literature in this area.

For the most part, these terms agree with terminology often used in systems engineering. In particular, it is important to note that a model is a physical, mathematical, or logical representation of a system, entity, phenomenon, or process. A simulation is an implementation of a model such that the behavior of the model, generally over time, can be observed. Also, simulation is a technique for testing, analysis, or training in which real-world systems are used or where a model of these systems reproduces real-world and conceptual systems. There are three different classes of simulations: constructive, virtual, and live. Constructive simulations are solely resident in software. Engineers attempting to conceptualize, design, and implement various facets of a product or process most often use constructive models. They have the benefit of being repeatable and generally fast and can be run stochastically and repetitively, thereby providing a means to quantitatively assess the inherent uncertainty in some tasks and processes. Virtual simulations have a constructive component but also explicitly include a human-in-the-loop component, although in an artificial setting. These simulations may also include "real" operational software intended to run in the fielded product or physical hardware end items. Although virtual simulation repeatability and consistency are generally suspect, it is very useful for human factors engineering and individual training purposes. Because there is a human in the loop, virtual simulations generally run in real time, which adds to their expense and limits the amount of stochastic data that can be generated within complex scenarios. Live simulations have human "players" complemented with a broad mix of constructive models and operational hardware and software and include a "realistic" simulation environment. Live simulations are generally very costly to conduct but are considered essential to validate operational concepts and tactics and for unit and combined arms training.

Simulation-based acquisition recognizes the increasing role that these computer-based simulations and synthetic environments have in designing for and validating the changed acquisition process and environment and the role that humans will play in this new environment. Acquisition reform covers a broad spectrum but is largely focused on using information technology (IT) to bring efficiencies and commercial practices to DoD acquisition. The acquisition reform website (DoD, 2000b) presents a number of useful and current documents concerning this subject.

Numerous studies have illustrated that the early stages of a program involving system definition are when most of a program's life-cycle costs are really determined, and the ensuing rush to build something quickly is inevitably followed by a lengthy period of design changes and modifications in order both to get the "system right" and, more importantly, to get the "right system." Even then, the initial build may be so far from the right system that no amount of modification can redress the initial flawed approach. It is for this reason that evolutionary acquisition approaches are a common contemporary suggestion.

We must understand that as we approach what appears to be almost limitless computing capacity and speed, the time consumed in running simulations is shrinking dramatically. This is particularly the case for those simulations absent any "human in the loop." The issue is less the speed of model or simulation execution and more whether appropriate models can be built for evolutionary acquisition. Reduced product development life-cycle times must also assure the objective of superior system effectiveness with well-understood and manageable costs. Models and simulations are being used today within almost every functional domain of system acquisition. Unfortunately, these tools reflect a broad spectrum of adequacy: Some are derived from historical data of no modern relevance, some are employed outside their intended realm, and many are using specific product data inconsistent with data used for the same product within another functional domain. Few of these tools, or their underlying data that support their use, are integrated or interoperable with each other. Thus, it is very difficult to capture total system effectiveness and cost or to facilitate integration of systems to result in a system of systems. Thus, we have a potentially huge computational capacity to support modern and rapid product development but without a systems design approach that harnesses this power.

It is necessary to ask whether industry will spontaneously arrive at an SBA environment compatible with the government's interest in the presence of government inactivity in this regard. All evidence suggests this is unlikely. It would require an investment of discretionary funds at a time when the market capitalization of major defense firms has fallen sharply. Also, an individual company's strategy will surely be to create competitive advantage for the company. Significant investments have been made and are continuing to be made in engineering tools for a variety of functions. The tools that a company purchases are mostly of commercial origin and are usually tailored to the specific need of the organization. This often results in creation of a unique product environment where, once a vendor and that product are established, it becomes very difficult to substitute a new supplier for one initially chosen.

The phenomena of *path dependence* and *lock-in* are particularly present in products and services based on IT innovations (Shapiro and Varian, 1999). These invert the usual return to scale notions found in conventional products. The defense industry, as well as automotive companies and other manufacturers, has begun to take note of its costs and dependencies associated with its information technology suppliers. Defense companies may choose to ignore this issue because the costs are passed along to the DoD customer and the situation has the added benefit of creating barriers against future competition for developed products. The situation becomes even more pronounced when tools that have no commercial counterpart are involved. This is clearly the case for most models of combat capability or military vulnerability. These tools, while often unclassified, are crucially linked to classified data that characterize specific threats or friendly system behavior. Because of the high degree of complexity of modern military systems, many of these tools and simulations are themselves highly detailed and sophisticated. Even if we ignore the technology and data classification issues, serious users of these tools are only found within the defense industry, and they may constitute an insufficient market for speculative investment by companies. Often, these very tools are essential during concept development and architecting. These are the tools that a prime contractor must employ in order to conduct comprehensive trade-offs of virtual designs and conceptual architectures.

Consider two choices that the government may have with regard to these tools, especially the situation where we become more reliant on virtual product demonstrations and evaluations potentially brought about by SBA. The first choice is to let each major prime contractor develop its own set of tools and simulations. This approach will likely result in each contractor investing significant funds on tools to capture the operational environment but where each set of tools reflects the contractor's proprietary view of the military mission under consideration. There will then be a need to get government agreement that the individual representations are valid and that they also faithfully represent performance, supportability, maintainability, and cost characteristics of the specific products that an individual vendor is proposing. The government is then faced with validating each contractor's tools and accrediting them for use in a specific source selection process. The government will also be placed in the position of having to compare and trade off each competing representation against all others in order to make a source selection decision. This is a formidable task, particularly in the absence of a predetermined strategy for how such source selections will be conducted and with only "virtual" product results available at the time of selection. This suggests a very program-centric approach toward model development. Each procurement will produce procurement-specific and perhaps service-specific models, model environments, and associated simulations. There will be little incentive to generate shared approaches to modeling complex environments, particularly with individual service-dominated views of the battle-space and associated operational requirements. This outcome will likely be costly and ultimately result in budget- and time-constrained tools reflecting mediocrity in their comprehensive understanding of evolving requirements.

Alternatively, consider an environment in which the government and industry are encouraged to jointly agree on the development of common models for individual mission areas. This suggests that the number of models and their purpose be managed to produce collaborative model development environments and simulations. Contractors would participate in model development and be afforded the opportunity to contribute model improvements, even as proprietary model objects where competitive issues are at stake. The government would specify how the models are to be used during source selection and a contractor would be required to "protest" in advance if it believed the evaluation model incorrectly captured the salient features of an anticipated product development proposal. This model environment would afford some level of interoperability with a contractor's indigenous IT-based tool set, achieved through interoperability standards and procedures. The overall tool environment would reflect a comprehensive strategy of how the government intended to interface to a contractor's environment, addressing both data and the interoperability of models as well as data that may reach beyond the specific procurement under consideration. This latter requirement would facilitate the evaluation of a "system of systems," "family of systems," or "federation of systems" (Sage and Cuppan, 2001) and the integration of data to conduct higher levels of aggregated analysis. Finally, the data requirements and formats would be made known to all contractors, and all data required for subsequent competition would be available.

The difficult part is to create a SBA environment, such as the one just described, that is based as much as possible on commercial tools and environments. The government should not want to stifle competition where a viable commercial tool environment exists; it needs to develop "world-class" approaches for the subset of tools for which there is no commercial market, ensuring that these tools are available to its suppliers and compatible with its internal environments.

Simulation-based acquisition calls for the virtual development of a system through iterative improvement of its model representations of the system, beginning with the identification of system concepts, continuing with the selection of "best" concepts and the evaluation of those concepts against user life-cycle requirements, progressing through manufacture and deployment, and ending with system retirement. As these myriad representations mature, test artifacts may be used to validate model descriptions and to reveal instances in which models do and do not properly represent *real world* conditions.

To “build” a comprehensive digital representation of a system whose authenticity is accepted by all interested parties is a daunting task. It requires cooperation among all stakeholders, and it also requires an environment that supports and encourages this level of cooperation on a large scale. Ideally, SBA will go a long way in helping to realize this cooperation by capitalizing on the synergy between a vastly improved culture, process, and systems engineering environment to enable people and organizations to accomplish work in an integrated fashion.

9.3 SIMULATION-BASED ACQUISITION: STRUCTURE, FUNCTION, AND PURPOSE

The Office of the Secretary of Defense (OSD) has expressed strong support for the concept of SBA. The DoD’s vision for SBA is “to have an acquisition process that is enabled by robust, collaborative use of simulation technology that is integrated across acquisition phases and programs. The purposeful objectives of SBA are to: reduce the time, resources, and risk associated with the acquisition process; increase the quality, military utility, and supportability of systems developed and fielded, and; enable integrated product and process development (IPPD) from requirements definition and initial concept development through testing, manufacturing, and fielding” (Sanders, 1997, p. 75).

Because SBA is an evolving concept, there are differing interpretations on its scope and method of implementation. In their book on SBA, Johnson et al. (1998) expanded the definition with a detailed explanation of a dominantly functional interpretation of SBA: “Simulation Based Acquisition is an iterative, integrated product and process approach to acquisition, using modeling and simulation, that enables the warfighting, resource allocation, and acquisition communities to fulfill the warfighter’s materiel needs, while maintaining Cost As an Independent Variable (CAIV) over the system’s entire life cycle and within the DoD’s system of systems.” The highlights of their definition are that “simulation based acquisition is . . . ”

- “ . . . ***an iterative, integrated product and process approach to acquisition***”—Thus, SBA enables IPPD teams, in which the DoD and contractor organizations work internally and with each other in an integrated team effort, to converge on trustworthy solutions through use of an iterative design process that is based on a well-adjusted set of system requirements.
- “ . . . ***through modeling and simulation***”—Modeling and simulation activities make SBA possible through creating a synthetic environment that enables exercising the power of simulation to explore many more iterations of virtual designs than would be possible with physical prototypes. The associated level of increased user involvement leads to better learning and problem solving than obtained from the more traditional approach obtained from physical prototypes. The resulting increased communication and enhanced learning make team members more effective.

- "*. . . **to enable the warfighting, resource allocation, and acquisition communities***" —A major objective of SBA is to integrate three principal acquisition support systems: the Requirements Generation System, the Planning Programming and Budgeting System (PPBS), and the Acquisition Management System (AMS) in support of the acquisition community and the related government and industry agents.
- "*. . . **to fulfill the warfighter's materiel needs while maintaining Cost As an Independent Variable (CAIV)***"—The desire here is to maximize need satisfaction to the maximum amount possible within resource constraints. The cost as an independent variable concept (Brady, 2001) will, in principle, allow more trustworthy predictions of the costs of different alternatives and thereby enable better informed analysis of trade-offs.
- "*. . . **over the system's entire life cycle***"—This suggests examination of all relevant facets associated with systems acquisition early in the acquisition life cycle and throughout acquisition of the system. These facets include the "ilities" associated with quality management of the acquisition process: affordability, availability, flexibility, interoperability, lethality, maintainability, manufacturability, mobility, reliability, supportability, survivability, and sustainability.
- "*. . . **and within the DoD's system of systems***"—This suggests investigation of all significant interactions within and across the various systems that, collectively, result in the overall system of systems. This should enable total systems integration and, ultimately, expansion of the system-of-systems concept to include federation-of-systems concepts (Krygiel, 1999; Carlock and Fenton, 2001; Sage and Cuppan, 2001) that are needed in combined operations brought about by collaborative allied systems and programs.

The above reference to a system of systems enables the capture of important realities brought about by the fact that modern defense systems are not monolithic. Rather, they have 5 characteristics (Maier, 1998) that makes the system of systems (Krygiel, 1999; Carlock and Fenton, 2001; Sage and Cuppan, 2001) designation most appropriate:

1. *Operational Independence of the Individual Systems* A system of systems is composed of systems that are independent and useful in their own right. If a system of systems is disassembled into the component systems, these component systems are capable of independently performing useful operations independently of one another.
2. *Managerial Independence of the Systems* The component systems not only can operate independently but also generally do operate independently to achieve an intended purpose. The component systems are generally individually acquired and integrated, and they maintain a continuing operational existence that is independent of the system of systems.
3. *Geographic Distribution* Geographic dispersion of component systems is often large. Often, these systems can readily exchange only information and knowledge with one another and not substantial quantities of physical mass or energy.
4. *Emergent Behavior* The system of systems performs functions and carries out purposes that do not reside in any component system. These behaviors are emergent properties of the entire system of systems and not the behavior of any component system. The principal purposes supporting the engineering of these systems is fulfilled by these emergent behaviors.

5. *Evolutionary Development* A system of systems is never fully formed or complete. Development of these systems is evolutionary over time and with structure, function, and purpose added, removed, and modified as experience with the system grows.

We see that the operational concepts needed for a trustworthy SBA process are not at all simple. There are many needed elements. There is a distributed data repository that contains all of the data about the product under development. This is centralized in a virtual sense, and all of the different stakeholders have a shared responsibility to keep the repository up to date such that all have rapid access, throughout the life cycle, to information required to understand and define, develop, and deploy a trustworthy system.

It is very important to have confidence and trust in the models and simulations that comprise an SBA approach. When questions regarding confidence in modeling and simulation activities are raised, most often this relates to the notions of verification, validation, and accreditation (VV&A) of models and simulations. Appropriate definitions of these terms may be found in a variety of sources (Banks, 1998; NDIA, 1999; U.S. Navy, 2001):

> *Verification* is the process of determining that a model or simulation implementation is transformed from one phase of development to another in a way that is consistent with the documented requirements and specifications. It is concerned with *building the model or simulation right*.
>
> *Validation* is the process of determining the degree to which a model or simulation is an accurate representation of the real world from the perspective of the intended uses of the model. It is concerned with determining that the model or simulation behaves with sufficient accuracy relative to intended purposes or with *building the right model or simulation.*
>
> *Accreditation* is the process of certification that a model or simulation is acceptable for use for a specific purpose. It represents official recognition that a *model or simulation produces credible results and is otherwise usable*.

In general, models and simulations are examined throughout the VV&A process from the users' application needs perspectives.

Many suggest that the SBA vision cannot be realized without an investment to develop the processes and architectures for the SBA way of doing business. They believe that specific strategies are needed to assure the appropriate level of data standardization and tool interoperability and that these strategies will not evolve spontaneously. Because of the continuing pressure on the defense budget, any suggestion of a new investment, no matter how small, comes under intense scrutiny within the Pentagon, and the funding to support SBA strategy development and execution generally requires a "business case" that will warrant the investment.

In developing this business case for SBA, it is necessary to understand the current state of product development and production. Models and simulations are today pervasive across all phases associated with engineering a system—definition, development, and deployment. One challenge is that tools used for modeling and simulation are generally not integrated and operate only on unique data that may be inconsistent across different views of the same product. Users of these tools can easily forget their limitations and may place unwarranted confidence in their results. A fully developed SBA environment will have integrated these models and addressed these concerns so that the procuring authority will, with confidence, understand the technical, design, cost, and operation performance risks of a product before any physical prototype of that product exists. This situation will be realized because the product will have been designed, tested, and operated in an integrated virtual environment that will, itself, be designed to illuminate the uncertainties of the integrated product knowledge as embodied in its interoperable models and simulations. Furthermore, modeling and simulation technologies will be applied to understand and project the trainability, maintainability, and supportability factors and costs for the equipment before it is produced or placed in the field.

Only when an SBA procuring authority has reached a satisfactory level of understanding of this virtual domain will it proceed into planning the next phases of prototype development, testing, and initial production. This planning will reflect a prototype and testing program that focuses on those issues in the virtual domain that revealed the weakest level of modeling and simulation "confidence," thereby demanding greater scrutiny before a final production decision. The resulting product prototyping and testing strategy should have, as a major objective, not just the validation of a point design, but the collection of sufficient data to improve the models and simulations, and should provide greater confidence in future virtual developments. Implicit in this process is that vendors will compete their designs in these virtual domains and the procuring authority will as a result have sufficient insight into both its own SBA environment as well as those of others in the supply chain in order to become and remain an informed procurer. This is particularly of significance when dealing with federated modeling and simulation issues (Nance, 1999; NDIA, 1999; U.S. Navy, 2001).

Even though the SBA concept is potentially appealing, there are a number of obstacles that need to be overcome. Much initial work must be accomplished before the first physical item is available using the SBA approach, especially in a distributed or federated environment. However, numerous studies have illustrated that the early stages of a program are when most of a program's life-cycle costs become determined. Premature cessation of the definition phase and proceeding to development with potentially volatile requirements and specifications in order to be able to build something quickly are inevitably followed by a lengthy period of design changes and modifications in order to obtain the right system. The initially configured requirements and specifications may be so flawed from appropriate ones that no amount of modification and associated expenditure can redress the initial flawed approach. Can we attach a cost savings to the solution of this problem through use of SBA? There is no shortage of anecdotal and factual data on the savings realized through modeling and simulation in defense acquisition. The Joint Simulation Based Acquisition Task Force (1998) has a 28-page discussion on SBA's *return on investment* (ROI) replete with examples of cost savings. The savings are impressive, but there may be difficulties in scaling the data up to the level of application envisioned for contemporary SBA processes and environments. The ROI calculations just attempt to do that, and one can indeed generate some numbers on the prospect of SBA that are so large they become very suspicious and apparently not believed.

While there is much discussion about SBA, there is very little in the way of formal guidance or a DoD-wide implementation plan for it at this time. Some have indicated that the SBA efforts are now the purview of the individual services and not the DoD and that it is not sufficiently emphasized in the new series 5000 regulations (Johnson, 2000). During the 1997 to 1999 time frame, industry attempted to describe the long-range vision for SBA. This was done by the SBA Industry Steering Group (SBA ISG) to the acquisition council, subordinate to the DoD Executive Council on Modeling and Simulation. Its ideas were developed in the ISG's SBA functional description document (FDD) (NDIA, 1999) and are summarized in the next few paragraphs.

The ISG believes that, when properly implemented, SBA can potentially make possible high-quality, enterprise-wide, collaborative decision making throughout the acquisition life cycle. Simulation-based acquisition is intended to be a process, culture, and environment whose use will result in more reliable and dependable assessments of the consequences of making acquisition decisions prior to funding commitments, thereby diminishing acquisition risk. This is to be accomplished by maximizing the use of relevant acquisition information while simplifying the process of capturing, managing, and assessing that information. In 1998, the ISG and OSD jointly declared the SBA vision as an Acquisition process in which the DoD and Industry are enabled by robust, collaborative use of simulation technology that is integrated across acquisition phases and programs. It was further stated that the goals of SBA are to substantially reduce time, resources, and risk associated with the entire acquisition process; increase the quality, military worth, and supportability of fielded systems while reducing total ownership costs throughout the total life cycle; and enable Integrated Product and Process Development (IPPD) across the entire acquisition life cycle. The apparent understanding here is that, as a new systems acquisition paradigm, SBA embraces the total system life cycle from initial realization of an unmet need, carrying all the way forward through system design production, operation, and retirement.

This paradigm is supported by three principal characteristics of the SBA process, as well stated in this FDD:

1. *SBA is an evolved culture* in which enterprise-wide and DoD-wide cooperation is the rule and individual technical contributions and innovations are encouraged and efficiently and effectively managed. This culture encourages needed changes, such as to lead to enhanced concurrent development and provision of incentives for organizations to provide tools and procedures for use by other organizations and without institutional or service-imposed barriers.
2. *SBA is a refined system acquisition process* that capitalizes on changes in the acquisition culture in order to facilitate collaboration by many integrated product teams (IPTs) across the entire system acquisition life cycle.
3. *SBA is associated with an advanced systems engineering environment* in which the application of various automated tools and methods supports all system life-cycle activities and encourages software reuse and interoperability maximization. This SBA environment provides a means to execute an extensible, tailorable, and repeatable acquisition process through creation of reusable product description repositories that can ultimately be used to cost effectively reengineer products for enhanced effectiveness and efficiency. This environment supports the seamless flow of data between acquisition, engineering, support, and training communities. This integrated SBA environment supports an evolutionary system acquisition process.

9.4 AN SBA APPROACH TO HUMAN SYSTEMS INTEGRATION

In the remainder of this chapter we will discuss approaches to be taken to achieve some of the benefits of SBA when addressing human systems integration (HSI) in contemporary acquisition environments. We will do so from the point of view of a "new" program.

Those concerned with integration of a legacy system as inherent in the new system being developed will need to tailor the suggestions made here in accordance with those systems integration needs that affect their program (Sage and Lynch, 1998). The approach presented here is based on experience in the DoD product development process and the precepts of SBA. It is written as a guideline for the HSI professional participating in the context of a large systems engineering development program.

In considering an SBA development strategy and related HSI concerns, it appears best to consider three different perspectives on process for engineering a system:

- program development objectives and related processes for engineering or acquiring systems;
- models, analysis, and data collection or methods and tools; and
- systems and program management.

9.4.1 Development Objectives and Processes

Texts on the topic of systems engineering and management (Sage, 1992, 1995; Sage and Rouse, 1999), devote a great deal of attention to the process of requirements definition, and it is the phase of the systems engineering life cycle for the ultimate product where virtual environments and simulations can have the biggest pay-off in the SBA context.

The requirements definition process is the first phase of any new program (including upgrade programs to existing equipment). For very large programs, such as acquisition category 1 (ACAT 1) programs, there will exist a mission needs statement (MNS), a capstone requirements document (CRD) generated by a commander in chief (CINC) of a unified command, and a supporting operational requirements document (ORD) generated by one of the military services. The derivation of these documents is an evolutionary process typically spanning a number of years. These requirements are very broad and are meant to be descriptive rather than prescriptive in nature. Nevertheless, they can have a dramatic impact on the human–system interface aspects of a design. For example, a top-level requirement of the navy's DD-21 program is to have a total crew complement of no more than 95 people, in contrast to the approximately 300 people normally found on a twentieth-century destroyer. Even if the HSI portion of the design was not exposed in any detail during the early requirements definition phase, it is important for systems engineers and systems managers to understand the source of the major system design requirements and how they came to be.

One must have a clear understanding of the operational purpose of the item being developed and the potential roles that HSI will play in various systems engineering design and development approaches that could satisfy operational needs. This understanding of operational requirements is fundamental to good system engineering practices. The next stage is to understand how different design concepts are to be assessed. It may be that there is either an explicit or implicit model of the system's value. This model could be expressed in terms of acquisition or operating cost or it could include a wide array of individual performance or cost metrics. In all likelihood, this model will address a large number of issues that involve, as well as some that are beyond, the purview of the HSI domain. It is important that we recognize this model, even if it is not explicitly expressed as a "model." Any set of requirements that is expressed in measurable design parameters is, in fact, equivalent to an assessment model for that system. If such a model is not explicitly stated, we must attempt to derive the model from implicit requirements. If the model exists or can be reasonably constructed from implied requirements, then our next task is to analyze the model from an HSI perspective and determine if it is appropriate. In some cases, the system requirements may be presented as unalterable, but that does not relieve us from establishing the model assessment priorities. As an HSI practitioner, the most important early step is to determine those aspects of the model that capture issues that are impacted by the HSI architecture of the solution.

We may find that there are no parameters in the model that appear to relate to the HSI "view" of the system. If this is the case, we have identified a potentially significant issue that needs to be addressed. In the "grand scheme" of things, if there are no HSI systems engineering parameters that impact the model that will evaluate the system, then a very strong argument exists that any investment in HSI is unnecessary. This is, of course, a foolish and untenable situation. If such a situation is detected, it will then be very desirable and necessary to establish the linkage between the evaluation model and the HSI parameters of the system that relate to the model in order to diagnose the issue and suggest potentially corrective measures. In most cases this should not be difficult. For example, if the probability of success is based on reaction to some external stimulus, then the issue is how the system to be engineered can decrease the reaction time, and this is often an HSI issue. If the linkage between the evaluation model and the human system design is not apparent, then it is up to the HSI designer to argue for the inclusion of the appropriate parameters. Failure to establish this connection must necessarily relegate the HSI part of the design to an insignificant part of the overall system design and related investment.

In addition to the assessment of the value of the HSI components of the system design, we must also consider the cost impact of HSI-related decisions. The question needs to be asked: Are these costs correctly portrayed in the cost estimating tools associated with the design? While we are collecting the needed information on the sources of the HSI requirements, we should also be developing an understanding of all the major system requirements and the technology and subsystem domains to which they have been allocated. In particular, we need to identify any domains that may impact or reflect HSI design decisions. We must identify the analysis and design environments in which major system requirements are to be assessed and whether their relationship to HSI parameters is properly handled. For example, in a situation where the performance of a system is highly dependent on a series of tasks in a platform with many workstations, the following questions are pertinent. Do other aspects of the design reflect assumptions about the performance of workstation tasks? How do they address uncertainty or human variability? Is there an associated risk that other parts of the design are assuming a best case approach and will be insensitive to subsequent changes in the HSI design as it evolves? One of the larger issues that SBA attempts to address is that of "harmonizing" and integrating different views of the system and to enable "real-time" incorporation of design changes and their implications across the entire design space. Until an IT environment is developed that can reliably perform that function, it is up to individual engineering teams to maintain this broad general awareness. They must constantly evaluate how their decisions relate to the performance requirements and objectives of the total system and the impact of these design decisions on domains outside the HSI field of regard.

9.4.2 Methods and Tools

While an effort is being made to understand the requirements world, we must also focus on the HSI engineering environment. In the SBA context this entails an evaluation of all of the

HSI data that are relevant to engineering the system and the tools, methods, and models that will be used to evaluate design trade-offs and analysis.

In this context, one might consider models in the manner described well by Blanchard and Fabrycky (1998, p. 91): A model is "a simplified representation of the real world which abstracts features of the situation relative to the problem being analyzed. It is a tool employed by an analyst to assess the likely consequences of various alternative courses of action being examined. The model must be adapted to the problem at hand and the output must be oriented to the selected evaluation criteria. The model, in itself, is not the decision maker but is a tool that provides the necessary data in a timely manner in support of the decision-making process. The extensiveness of the model will depend on the nature of the problem, the number of variables, input parameter relationships, number of alternatives being evaluated, and the complexity of operation. The ultimate objective in the selection and development of a model is simplicity and usefulness." The authors suggest that models should represent the dynamics of the system in a way simple enough to understand and use and close enough to reality to yield successful results; highlight those factors that are most relevant to the situation at hand and repress unimportant ones; be comprehensive through inclusion of all relevant factors; be reliable in terms of repeatability of results; be simple enough to allow timely implementation and use; and incorporate provisions for ease of modification or expansion to permit evaluation of additional factors that are not immediately apparent and that occur later. These are not necessarily trivial features to incorporate in a model. This suggests that models themselves should be adaptive and evolutionary.

In the SBA context, models of the system's behavior and the data describing the instantiation of the product being evaluated by those models represent the system description at that point in its evolution, wherever that point may be in the life cycle of engineering the system. Indeed, the adequacy of the models in confidently predicting the consequences and behavior of the system is a direct reflection of the overall risk of the conceptual system design itself. Any difficulty in modeling a system should be cause for a reevaluation of our own understanding of the factors that are relevant to its description. There is a general precept in the world of SBA that if we cannot model a system's behavior and interactions, then we have a poorly understood basis for engineering the system.

A strong word of caution should be injected at this point. When confronted with the challenge to produce an adequate model of a system, the immediate reaction is often to "go off and create one." This is often an inappropriate reaction. There are a lot of models that may potentially be used, so the first task should be a thorough survey and understanding of what is available. With that knowledge, we next should determine what is actually usable and appropriate in the situation at hand. Very often the best situation is one where a model has been developed and is available commercially. Such models may well be very responsive to requirements and are almost always cheaper than developing an in-house approach. Ultimately, we will have to make the choice about what models to use "off the shelf" and what to develop. However, developing a model from scratch carries a heavy burden of VV&A, as described earlier. Self-developed models should deservedly be met with customer skepticism until model accuracy and appropriateness have been fully demonstrated. Even if these conditions are met, engineering models are rarely static; they require a steady diet of funding since they need to evolve and maintain currency with the underlying technologies that they are to emulate.

Development of the SBA concept requires strong focus on the need to share data among different models and simulations. Ultimately, this may include real-time interaction among detailed engineering simulations and cost models operating “synchronously.” This is an area that has the potential to yield great insight, but it will also doubtlessly result in a great deal of confusion as we try to make models work together through integration. Once we have the models identified, they can be used to analyze the system in a great variety of ways, some of which were not possible before the era of computer-implemented mathematical models that can be run at very great speed to enable experimentation with potentially complex adaptive behavior. Future models will be exercised over a very broad range of parameters that otherwise might remain unexplored.

Within any domain such as HSI or any subdomain, there may be many models and simulations that can be brought to bear, but we very often find that the use of these models is limited by the availability or reliability of data. Furthermore, when an engineering effort within a domain completes an analysis and creates new data relevant to the design, it is very often not clear how the data are distributed to the affected parties outside that engineering domain. There are important data interpretation and configuration management issues that must be addressed for SBA to be implemented successfully. Generally, the data problems for the SBA environment are of two types: (1) understanding the data as information and knowledge and (2) distributing and managing the data. Understanding the data refers to the need to share data with others, who may not know very much about the source of the data or their correct interpretation. This is a nontrivial problem when large systems are being analyzed, compared, and engineered. This often has to do with the underlying assumptions or conditions that existed at the time the data were created. Today, we often find ourselves with apparently useful data to address a question but with little confidence about some of the underlying attributes of the data. This is often because different engineering communities, technology domains, or cultures assume different things and terms have different meanings. In a world where the communication across technology domains and engineering teams is strictly controlled, the interfaces across boundaries can be managed to minimize this source of communication problem. However, SBA envisions an engineering environment where there are few boundaries and information can flow effortlessly and instantaneously across the engineering enterprise. Thus, there is a need for a more organized way to retain all of the important information about a data file so that others can properly interpret it. This is typically accomplished through the process of “data modeling,” another growing field in the area of IT.

Data modeling attempts to create an unambiguous description of data and the relationships among data elements. Often this data model is “object flavored,” and relationships between all data entities are mapped (e.g., parent–child relationships) and the attributes of each data entity are explicitly defined. Communities of interest have begun to develop their own data models to be published as international standards. For example, manufacturers have been developing standards for the exchange of parts data under the auspices of a global organization whose sole purpose is to establish standards, such as the International Organization for Standardization (ISO). There are many domains where there are no definitive terms and conventions for creating and managing data. Among those fields with no definitive data model are cost estimating and analysis and human–system interface design. Some aspects of human–system interfaces may already be accommodated by related engineering standards efforts, but data dealing with human system performance may have no common format or well-understood interpretation beyond the realm of a small subcommunity of practitioners. Setting data standards is not a panacea, and in some engineering domains and areas of research it may be premature or inappropriate. Furthermore, establishing and maintaining international standards can be a slow process.

Also, a difficulty with standards is that there are often so many from which to choose. Despite these shortcomings, the HSI community desirous of actively participating in an SBA program environment must come to grips with how data about the HSI aspects of a design are collected, stored, and shared. At a minimum, the HSI effort should include development of a data dictionary. This is an explicit definition of all the terms the team is going to use and apply to HSI data. Ideally it would also specify all of the attributes about a data file, model, or simulation that are considered essential and must be retained with that item. As discussed earlier, we must also identify all of those domains that will be providing data and those that will be recipients of HSI data. It is then necessary to understand all the issues about format, assumptions, and constraints on the data that apply. The significant benefit of a data model is that it can make all of these issues explicit and immediately visible to someone who is searching for or viewing the data. It may also be possible that a program could create its own data model, integrating individual pieces from the ISO or U.S. standards-setting bodies [e.g., American National Standards Institute (ANSI), Electronic Industries Association (EIA), and Institute of Electrical and Electronics Engineers (IEEE)]. This program-specific data model may be the best strategy to ensure existence of a data-modeling approach that can be shared across all programs.

Some may ask: “Why is it necessary to create a data model for a program? If we have to share data among two domains or applications, we will simply write a translator.” This is indeed how many companies have been handling their data exchange needs. But that approach is becoming very costly, because it creates an *n*-squared problem. That is, if we have a relatively small program that uses 100 different applications that share different pieces of data (e.g., cost analysis spreadsheets, schedules, technical data, analysis results, simulation results, stored in spreadsheets, data files, text files, etc.), then we would need to write a data translator for each pair that needed to exchange information. Even if each application only exchanges data with 10 other programs, we would need 1000 translators or 500 bidirectional exchange translators. Furthermore, any time that an application changed, 10 translators would have to be inspected and potentially modified. The translation problem explodes exponentially for very large complex programs that may be sharing data between, e.g., contractors or clients/customers. If a virtual repository exists under one data model, then this translation/inspection only happens once for each application.

Another major issue about data in the SBA paradigm is that of configuration management (CM). Sage (1992) defines CM as the systems management process that identifies needed functional characteristics of a system early in the life cycle, controls changes to those characteristics in a planned manner, and documents system changes and implementation status. Determination and documentation of who made what changes, why the changes were made, and when the changes were made are the functional products of CM. Under SBA, the CM function must be maintained on all data, models, and simulations that impact a system or are used in the acquisition life cycle.

We will not expand this further here because CM is a well-recognized concern for programs and will only emphasize that it is even more important in the SBA context. The CM of the data about a product and the configuration of the models and simulations themselves are all very important SBA ingredients. This can become a complex problem when the effort is carrying forward multiple alternatives, each with its own data and each having unique performance characteristics predicted by the modeling and simulation environments. The CM system not only must keep track of data associated with each design but also should be able to track the configuration of a tool or simulation that produced the data or supported a specific design analysis. This is important because the broad sharing of data will only be successful if users of the data have confidence in its source and its "pedigree." Users must be able to look into a model or data file and learn all they need to about that data in order to determine that it is appropriate for use in some other application. Early in this chapter we introduced the notion of VV&A. The CM of data and models must include the VV&A attributes (e.g., date and source) of those items. Finally, we should remark that CM issues in an evolutionary acquisition context have yet to receive definitive study and consideration.

Once we have established how data can be understood and interpreted, you will have taken a major step in enabling the sharing of data across a program. The actual distribution of the data is more of a classical IT problem with many different approaches. Simulation-based acquisition simply recognizes that when models and data are going to be shared on such a broad scale, the underlying data repository and distribution system must appreciate the sophistication of the SBA paradigm. The most pressing requirement is that of data, and information, control and access. The fundamental goal of SBA is to achieve faster and more effective product development and support through the rapid exchange, understanding, and exploitation of all the data and information that exist about a product. But there may be valid reasons for restricting this information flow based on national security issues, company proprietary concerns, competition sensitivity, and other relevant factors. The IT communication and data backbone must satisfy these concerns.

9.4.3 Systems and Program Management

It is not difficult to take the position that implementing SBA suggests that we are "systems engineering" the acquisition process and the program management process from the perspective of creating a modeling and simulation environment that optimizes managerial effectiveness and problem solving. The models and simulations that we are referring to cover not just the domains of the engineering of systems but also the management functions of technical direction, planning, scheduling, and virtually all the tasks that are carried out under the systems engineering and systems management umbrella.

Topics of major interest for system engineering and management are the scope and type of development model implied by the evolutionary acquisition concept and the potential use of modeling and simulation in achieving this. The scope of SBA covers all the phases of a system's life cycle. Itemizing the list of subdomains based on the perspective of the user, owner, or builder would produce a lengthy list. Figure 9.1 presents an evolutionary life cycle that was comprised of three phases: definition, development, and deployment, and this life cycle can easily be expanded to yield a more realistic number of phases, such as shown in Figure 9.2. This figure represents a single build in Figure 9.1 as expanded into three life cycles: research, development, test, and evaluation (RDT&E); systems acquisition; and planning and marketing. A realistic systems engineering acquisition life cycle is necessarily associated with a life cycle for planning and marketing and a life cycle for RDT&E. In Figure 9.2, the life cycle for acquisition is expanded from the basic three phases of definition, development, and deployment to a more realistic seven-phase life cycle. Discussions of these expanded systems engineering life cycles and such related concerns as risk management may be found elsewhere (Sage, 1992, 1995; Sage and Rouse, 1999).

When we consider the engineering of a system, we also often find ourselves considering architectural views or perspectives. Many discussions of systems architectures

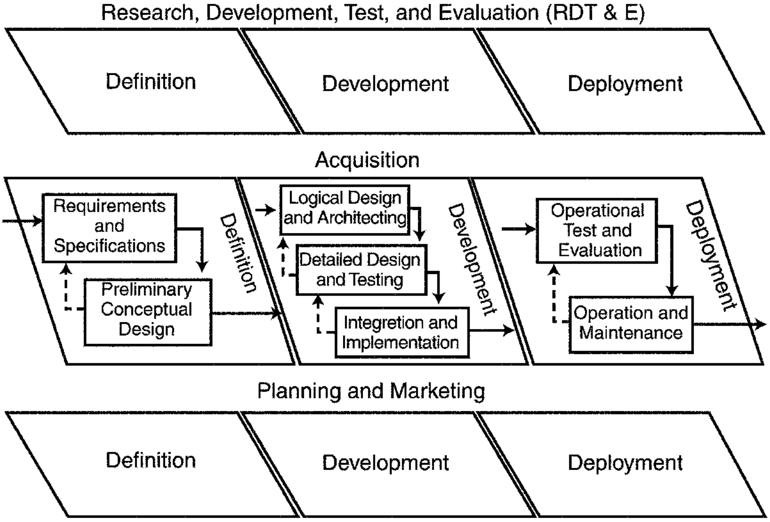

Figure 9.2 Research, development, test, and evaluation.

focus on three primary architectural views. Here we will use *functional*, *physical*, and *implementation* to describe these (Sage and Lynch, 1998). The approximate corresponding DoD terminology in the Joint Technical Architecture (Sage and Lynch, 1998) is *operational*, *systems*, and *technical*[1]. Development of the implementation or technical architecture is the process during which the entire physical system design is integrated. This process also provides the raw materials for definition of the system's external and internal interfaces. Each of these activities in the design process is first completed at a high level of abstraction and correspondingly low level of detail. This results in an initial implementation or technical architecture for the system at a high level of abstraction. Then the entire process is repeated at a lower level of abstraction associated with greater detail for the next level of components. This repetition at lower and lower levels of abstraction and greater and greater detail is continued until ultimately the detailed implementation architecture is realized. The associated decisions and designs are reviewed, and changes are implemented at the higher levels of abstraction to the extent needed and then iterated downward. The implementation architecture integrates system requirements with the functional and physical architectures. This process also provides the raw materials for definition of the system's external and internal interfaces. These three architectures are first conceptualized at a high level of abstraction and correspondingly low level of detail. This first results in an implementation architecture for the system at a high level of abstraction. Then the entire process is repeated at a lower level of abstraction associated with greater detail for the next level of components. This repetition at lower and lower levels of abstraction and greater and greater detail is continued until ultimately an implementation architecture is realized. The associated decisions and designs are reviewed and changes are implemented at the higher levels of abstraction to the extent needed. Sage and Lynch (1998) describe a multistroke decomposition process for architecting a system and its interfaces that is roughly equivalent to this.

We have emphasized that systems engineering is a multiphase process. Each of these phases can be viewed at a number of levels: family of systems, system, subsystem, component, and part. These are generally defined in a functional block diagram structure for the system being engineered. At each of these levels, the various phases of the systems engineering process need to be enabled through identification of appropriate various work efforts. A work breakdown structure (WBS) or system breakdown structure (SBS) is an appropriate way to display this information. We may identify a two-dimensional matrix framework representation of the phases and levels in the form of hierarchical levels in the SBS, as shown in Figure 9.3. When we recall that this framework needs to extend across each of the three major systems engineering life cycles and the family of systems may be comprised of a large number of "systems," the complexity of the effort to engineer a system becomes apparent. This provides major encouragement for modeling and simulation as a part of the effort to successfully engineer a system.

A challenge for systems management is to determine how the investment in the infrastructure of data, models, and simulations is developed and evolved. There are no easy answers here, and it will be up to each program manager to determine the best strategy for a particular program. It is perhaps most important to begin with an open mind and the approach that many take toward quality. Neither quality nor a productive SBA environment is "free," but both bring the potential for far greater success in the long run if properly addressed and managed.

Relative to systems management, it is also important to transform the engineering and acquisition cultures in order to be able to accept the broad sharing of tools and data that are implicit in SBA. This has been suggested as the most important factor in implementing SBA. The need for cultural change focuses on the need to share data across different domains of the acquisition process, so that agents focusing on different modalities of the same design are using the same or consistent data. A fundamental objective is the appropriate and early involvement of stakeholders that today exist at the periphery of the acquisition process. This includes agents involved in the training and maintenance of a system as well as other systems with which the primary system must interoperate. The method of participation of these agents within SBA would be through models and simulations that portray the diverse key interests and unique cost and performance sensitivities appropriate to the system architecture and design.

Phase Level	Definition	Development	Deployment
Family	Artifact$_{11}$	Artifact$_{12}$	Artifact$_{13}$
System	Artifact$_{21}$	Artifact$_{22}$	Artifact$_{23}$
Subsystem	Artifact$_{31}$	Artifact$_{32}$	Artifact$_{33}$
Component	Artifact$_{41}$	Artifact$_{42}$	Artifact$_{43}$
Part	Artifact$_{51}$	Artifact$_{52}$	Artifact$_{53}$

Figure 9.3 Framework for activities by level and phase.

There are many impediments to the timely exchange of data and models to conduct these kinds of early trades, including issues of job security and the fundamental fear that someone else may use “my” data inappropriately. An example of this threat to current business practices is the impact on product testing during development and operational evaluation. Simulation-based acquisition suggests that as much testing as possible should be conducted in the virtual domain and a physical item test should only be scheduled after thorough evaluation of the shortcomings of models and simulations to address the risks being addressed by the physical test. Also, a major objective of any physical test should be to improve the models and simulations of the test parameters such as to reduce the number of future tests. Test and evaluation professionals need to understand, accept, and evolve this concept.

Cultural barriers include issues associated with sharing data between customers and suppliers and between teams competing relative to new opportunities. Whatever the environment for sharing, it must provide appropriate safeguards for the protection of proprietary information; however, it should not unnecessarily restrict the flow of data. The SBA initiative to date has largely been supported by those who have had a historically strong role in the evolution of modeling and simulation (M&S), specifically those supporting the development of simulators for training and wargaming and simulations used in performance trade-offs at the conceptual phase of product engineering. A major cultural challenge is to educate the engineering and support organizations that SBA is not just a classical M&S “fad,” but a true initiative that requires that the broader engineering and management constituents of the acquisition process become major contributors and leaders of SBA practices. Thus, SBA can in no way be regarded as a replacement for systems engineering and management; it is an enhancer of good systems engineering and management efforts. It allows for wide-scope display of information and knowledge and thus supports the development of learning organizations. It does not represent a loss of responsibility and accountability or of security and competitive advantage. Rather, it is intended to enhance these for the betterment of all.

There have been a number of efforts to implement strong modeling and simulation capabilities in support of system acquisition. Particularly noteworthy among these is the U.S. Army (2001) Program for Simulation and Modeling for Acquisition Requirements, and Training (SMART). The Army Model and Simulation Office (AMSO, 2001) provides institutional support for SMART as the U.S. Army initiative that promotes the robust use of M&S efforts integrated across acquisition programs in an effort to reduce total ownership costs (TOCs), provide quicker delivery of products to the field, and simultaneously increase utility and worth of engineered systems. SMART is intended to more closely integrate the efforts of the requirements, acquisition, and training communities through the use of a variety of modeling and simulation approaches, including SBA. SMART is intended to foster collaboration across these three communities by integrating M&S beginning at the earliest phases in the acquisition process, thereby allowing better understanding of the process and enhancing its productivity and trustworthiness. SMART involves rapid prototyping to facilitate systems engineering so that the ultimately deployed systems meet users’ needs in an affordable and timely manner with minimal and controlled risks. The intent is to enable collaborative environments across organizational and functional barriers among users, developers, testers, sustainers, and trainers. Analysis of alternatives (AOA) and CAIV are two of the analyses that support the decision process early in the life cycle.

SMART initiatives require that a comprehensive management and technical strategy for HSI be initiated early in the acquisition process in order to ensure that human performance factors are considered throughout the evolution of the system design. SMART requires that human factors engineering requirements be established in order to develop effective human–machine interfaces and to avoid system features that require extensive cognitive, physical, or sensory skills. Also, it requires that systems be designed for human interaction to minimize human errors in using deployed systems. Various M&S tools are suggested to support decisions and trade-offs through analysis of design suitability and prediction of the effects of alternative designs and architectures on human–system effectiveness. Two authoritative publications describe the current state of this comprehensive effort (U.S. Army, 2000, 2001), and the latter of these contains a comprehensive listing of resources, including websites, relating to M&S for system acquisition.

9.5 SBA QUALITY ASSURANCE QUESTIONS

Today, many smaller DoD programs are significant users of M&S technologies. Because of this, many assert that they are already executing SBA. In some cases they are motivated to take this position because they believe that SBA is the "current buzzword" and by asserting an SBA capability they will win both status and funding. We should be careful about how we view this position.

Programs that are effectively employing M&S are likely to be the biggest proponents of SBA. They are the "believers" and provide the demonstrations that the use of M&S is saving money and reducing risk for their programs. For these reasons, SBA advocates do not want to alienate these aggressive M&S adopters. But the aggressive use of M&S simply represents the evolutionary path to SBA, not the aggressive goals of the SBA vision, and we need to make this distinction. Program-specific M&S adopters will agree that they do not have the ability to readily exchange data across M&S environments. They also feel no commitment, because they have no associated funding, to support other programs or develop models or simulations with broader application than their own immediate needs. Finally, none of these programs set out with SBA as the kernel of their acquisition strategy and therefore have not invested in the infrastructure required to optimize the benefits of SBA.

But the revolution to SBA cannot be just for brand-new programs, and it cannot wait until the full infrastructure is in place. What should a program be doing while it is waiting for the revolution, and how will it know how well it is doing? With these questions in mind, we have prepared a *very preliminary draft* set of quality assurance questions that might ultimately be used for an SBA capability maturity model (CMM) assessment checklist based upon SBA desiderata established in a report of the Joint Simulation Based Acquisition Task Force (1998) and other readings. This SBA quality assurance checklist may help a program manager better understand where a program is and whether it is creating the opportunity to reduce program cost and risk through incremental implementation of the SBA vision. This checklist does not conform to the formal staged structure of the Software Engineering Institute's systems or software CMM or any of the related efforts, such as capability maturity model integration (CMMI) initiatives, with their progressive levels of maturity that may happen later. For the moment it is simply a set of questions that might be asked by a government program manager:

1. *Collaborative Environment* (*CE*) Is the program participating in a multiproduct CE in which the exchange of data is facilitated through standards and configuration management? What is the purpose of this collaborative environment? Are tools being shared?

2. *Distributed Product Description* (*DPD*) Do you have a description of your product that is maintained as the "virtual baseline" for your product? Is it used as the primary reference for design, development, and analysis? Is it under configuration management? Is it responsive to the needs of the collaborative environment? Has this distributed information been integrated such that it appears to users as a single integrated information repository and is this DPD consistent and coherent and with sufficient access control to protect classified and proprietary information?

3. *M&S Planning* Does the program have an M&S plan? This plan should identify the full spectrum of program M&S constituents. It should have a plan for prioritization of those models and simulations that will bring greatest benefit to the program, both individually and if their data are shared and interoperable. Does the plan address the collaborative use of M&S and how the CE will be supported? Have M&S shortfalls been prioritized? Does the M&S plan reflect an investment strategy for models and simulation purchases or upgrades that will bring the greatest cost benefit to the program? Does the plan provide explicit guidelines for VV&A for models and simulations? Does the plan address configuration management of M&S used within the program? Has the M&S plan been integrated into the system engineering management plan (SEMP)?

4. *Program Management* Is the potential of M&S fully identified and addressed in the program management plan (PMP)? Areas to look into include cost-estimating tools and decision analysis tools. The PMP should also have a statement regarding the certification and credibility of models to be used. It should reflect an understanding of the importance of VV&A for models that may have a key impact on program decisions. Does the PMP address the significance of configuration management of the tools and data employed by the program? Does the source selection plan address the use of M&S? Will all tools used in source selection be provided to potential offerors?

5. *System Engineering and Management* Is the potential of modeling and simulation fully identified and addressed in the SEMP. Were M&S tools used in the system requirements analysis? Are those tools in use today? Does the SEMP require a specific evaluation of the use of models or simulations to assess performance, reduce cost, and minimize program risk by all functional engineering domains? Are appropriate tools integrated and/or do they generate and use data interoperably with other key M&S elements in the program? Is the DPD used as the source/repository for all data on the current state of the baseline design and design alternatives? Does the program have an integrated approach to data management to support the system engineering process? Is this approach further integrated into the program information management system?

6. *Test and Evaluation* Is the potential of M&S fully identified and addressed in the test and evaluation management plan (TEMP)? Will M&S be used to identify the highest priority physical tests, i.e., those necessary to assess the most critical parameters for which M&S tools are inadequate? For all physical tests that will be required, has a specific analysis on why models or simulations cannot be substituted been conducted? Have anticipated physical test parameters and results been projected through models and simulations? If not, what result is expected from the test, and what is the basis for these expectations? Will test events generate information to upgrade/improve existing models or simulations? Will physical tests provide statistically relevant sample sizes? Will M&S be used to interpolate or extrapolate physical test results? Are the methods valid?

7. *Data Management* Does the program have an approach for participating in the appropriate integrated data environment (IDE) or for establishing a new IDE if necessary? Does the IDE reflect an integrated data-sharing approach for all engineering functional and life-cycle domains? Does the IDE identify who are the users and suppliers of data and how these data are to be shared? Will the IDE enable interoperation of tools and methods? Does the IDE reflect a plan for continuous data object/element ownership?

8. *Knowledge Management* Does the program have appropriate plans to enhance knowledge generation, transfer, and sharing? Does this lead to efficient and effective knowledge sharing across all elements of the DPD and among all concerned parties such that they have a mutually consistent and accurate understanding of the system, system of systems, and family of systems to be acquired.

9. *Technical Performance* Has the program identified and prioritized the technical performance measures (TPMs) for the program? (TPMs are those quantitative factors that represent the most important product features in the eye of the customer.) What analysis methods were applied in the requirements analysis to derive these TPMs? If models or simulations were used, are these tools still appropriate and are they being used to validate the current state of the design? Have these tools evolved as the product has evolved? Are other models or simulations being exploited to understand and mitigate performance risk and program cost associated with these TPMs?

10. *Cost Estimating and Analysis* How does the program assess cost across three domains: (1) cost of current program execution, (2) projected recurring cost of the prime items to be delivered, and (3) life-cycle cost of the complete system? What cost models are employed and how effective are these models at projecting cost? Are they integrated? The program manager must appreciate the importance of understanding life-cycle cost and its impact on system design. There are few programs today that do not have a major objective of minimizing TOC. Because we cannot collect historical cost on future systems, we have no choice but to use a model or simulation to estimate future costs. Are the tools the right tools? Can they assess CAIV? The program manager must appreciate that many cost tools use historical data and are therefore limited in their ability to project cost benefits of technology improvements and state-of-the-art changes in processes and methods. How do the program's cost models handle this? Do these models maintain currency with the existing design? (That is, are they synchronized with the product development process?) Does the customer understand and have confidence in the suggested cost-estimating methods?

11. *Source Selection* Will models or simulations be used to evaluate competitive offers? What tools will be used? Do the competitors have these models? If not, why not? Do the potential offerors trust these models? What is the plan for how contractors can propose model improvements and modifications to better reflect their potential offer? Is proprietary information safeguarded?

12. *Contractor Use of M&S* Does the program office staff have sufficient knowledge of those models employed by the contractor that can have a direct impact on the government's assessment of overall contractor performance? If not, how will the government measure and assess the contractor's performance under the contract?

13. *Program Schedule* Do the program schedule and major milestone events reflect key demonstrations of M&S maturity and capability?

14. *Customer Satisfaction* Does the customer understand the value of M&S in reducing cost and understanding and mitigating risk? Is the customer comfortable with the use of M&S? Do you understand where he or she is uncomfortable and do you have a plan to address the issues?

15. *Evolutionary Development* Have the systems engineering life-cycle process and the systems management efforts been configured to support evolutionary acquisition? Is the CM and configuration control board structured so as to cope with the various facets of evolutionary acquisition? Have appropriate modeling and support tools and processes for evolutionary acquisition been obtained and does the culture of the organization(s) support use of these?

16. *Human Systems Integration* Have appropriate planning and action been taken to ensure that relevant HSI concerns are addressed throughout the acquisition effort?

9.6 CONCLUSION

Simulation-based acquisition in some form is inevitable. It is simply the logical progression of the process with which we select and develop products to satisfy future needs. It is not at all merely using M&S to support systems acquisition. Highly sophisticated applications of M&S techniques already exist in the aerospace, automotive, and heavy-equipment industries. These provide demonstrations that product simulation and experimentation are powerful concepts. The SBA approach may be the only way that large, complex, and costly systems can be developed and tested before committing to production. Fundamentally, SBA is a risk mitigation strategy. It recognizes that all that we know can be modeled and what we do not know may be the best justification for intense model development as a method to focus our identification, pursuit, and resolution of the unknown risks.

Simulation-based acquisition has fundamental and profound implications for system engineering. System engineering in the SBA environment must be the focal point for architecting and creating the simulation paradigm of a system development and life-cycle support activity. But no one doubts that there is plenty of opportunity to spend money in the wrong places. Since SBA is a concept with much to be filled in, it offers the opportunity for invention and innovation in how we get there. It is a chance to “system engineer” the acquisition and product development process itself.

In closing this chapter, the authors recognize that we present an optimistic view of the potential of SBA to reform and improve the system engineering and acquisition environment. However, we recognize that there are at least three areas of concern that we have not dwelt on and that deserve far greater treatment that is beyond the scope of this chapter. These three areas are (1) commercial analogues to SBA, (2) simulation of combat or stressful environments, and (3) the shortcomings or “gaming” of models that misrepresent a system or preserve a point of view.

In this chapter we have emphasized SBA as an initiative within OSD, because that is where the abbreviation comes from and that is where it is being applied. The notion of SBA arising from the DoD is not inconsistent with the observation that, historically, many management and technology innovations that have broad commercial potential and utility have their origins in “state-of-the art” defense programs. That notwithstanding, there is a broad appreciation of the benefits of M&S in commercial industries. This is particularly true among vehicle manufacturers. For example, auto manufacturers are placing increasing reliance on M&S to assess driver comfort and dashboard design as well as precursors to crash tests and the assessment and design for occupant safety. The details of their approaches are worthy of more detailed investigation. We should point out that a big difference between DoD and commercial ventures in the implementation of SBA lies in the fact that most large-scale U.S. manufacturers have pervasive control over their processes. If they see the economic benefit to some aspect of SBA, they can generally change quickly. The DoD and the defense industry operate under a more pervasive set of constraints, limiting the ability to both initiate and fund process changes.

Perhaps the single greatest shortfall in the world of DoD simulations is the ability to model human performance under stress. This includes speed and accuracy at the console as well as the effectiveness of troops under fire and experiencing casualties. The authors know of no easy solutions here. Continued investigation is obviously required, but there may be a point where the best we can hope for is to bound the problem within some range of anticipated performance based on training, leadership, motivation, and other factors, which, admittedly, are rather difficult things to quantify.

The final issue is the inadequacy of some models or even the blatant misuse of models and simulations to protect a system or position. This is a real-world problem, often brought about because the best representation of a system's performance is often that provided by the program management office responsible for fielding the system. Indeed, because of the way money flows in the acquisition process, the program office may be the only source of funding for the construction of the system model/representations. This is a potentially very dangerous situation. The danger is greatest for sophisticated or complex systems that require significant understanding and model detail to capture their behavior. One of the early reviewers of this chapter pointed out the failure to adequately predict weapon susceptibility to enemy countermeasures. The authors believe that the failure is due less to diligent understanding of the system's shortcomings and more to the failure to have an independent and qualified team objectively assess and model the system. In any case, the fact that we can model a system behavior with great fidelity does not provide the guarantee that we will. This is one of those process and cultural issues that must be focused on in the evolution of SBA constructs. These are continuing concerns in the effort to fully develop simulation-based acquisition in the 21st Century (National Research Council, 2002).

NOTE

1. Some authors, such as Buede (2000) use the term *operational architecture* to describe essentially what we call the *implementation architecture*. Either is an acceptable term. We use the term implementation primarily because it is the term used in the National Institute of Standards and Technology (NIST) systems integration and management architecture (SIMA) and to avoid possible confusion with the DoD joint technical architecture, where the term *operational* is used with a somewhat different meaning.

REFERENCES

Acquisition Functional Working Group. (1999, December). *Acquisition Modeling and Simulation Comprehensive Core Body of Knowledge*. Department of Defense. Available: www.dmso.mil.

Army Modeling and Simulation Office (AMSO). (2001, April). *Simulation and Modeling for Acquisition, Requirements and Training (SMART) Reference Guide*. Available: www.amso.army.mil/smart/documents/ref-guide.

Banks, J. (Ed.). (1998). *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice.* New York: Wiley.

Blanchard, B. S., and Fabrycky, W. J. (1998). *Systems Engineering and Analysis*, 3rd ed. Upper Saddle River NJ: Prentice-Hall.

Boehm, B., and Hansen, W. J. (2001, May). The Spiral Model as a Tool for Evolutionary Acquisition. *Crosstalk*, *14*(5), 4–11.

Brady, J. H. (2001). Systems Engineering and Cost as an Independent Variable. *Systems Engineering*, *4*(2), 233–241.

Buede, D. M. (2000). *The Engineering Design of Systems*. New York: Wiley.

Carlock, P. G., and Fenton, R. E. (2001). Systems of Systems (SoS) Enterprise Systems Engineering for Information-Intensive Organizations. *Systems Engineering*, *4*(4), 242–261.

Defense Systems Management College. (1998). *Joint Logistics Commanders Guidance for Use of Evolutionary Acquisition Strategy to Acquire Weapons Systems*. Fort Belvoir, VA: Defence Systems Management College Press. Available: www.dsmc.dsm.mil/pubs/gdbks/jlc_evol_acqstrat.htm.

Johnson, M. V. R. Sr., McKeon, M. F., and Szanto, T. R. (1998, December). *Simulation Based Acquisition: A New Approach.* Fort Belvoir, VA: Defense Systems Management College Press. Available: www.dsmc.dsm.mil/pubs/mfrpts/mrfr%5F1998.htm.

Johnson, M. W. (2000, June). Big Industry View of SBA. Paper presented at the National Defense Planning and Analysis Society (DPAAS) Conference on Simulation Based Acquisition. Available: www.dpaas.com/SBA%20M%20JOHNSON.PDF.

Joint Simulation Based Acquisition Task Force. (1998, December). *A Road Map for Simulation Based Acquisition*. Available: www.acq-ref.navy.mil and www.msosa.dmso.mil/sba/documents.asp.

Krygiel, A. J. (1999). *Behind the Wizard's Curtain: An Integration Environment for a System of Systems*. Washington DC, Command and Control Research Press.

Maier, M. W. (1998). Architecting Principles for Systems-of-Systems. *Systems Engineering*, *1*(4), 267–284.

Nance, R. E. (1999). Distributed Simulation with Federated Models: Expectations, Realizations, and Limitations. In *Proceedings of the 1999 Winter Simulation Conference*, Society for Computer Simulation International, San Diego, CA (pp. 1026–1031).

National Defense Industry Association (NDIA) SBA Industry Steering Group. (1999, February). *Simulation Based Acquisition Functional Description Document*. Available: www.msiac.dmso.mil/sba/documents.asp.

National Research Council, *Modeling and Simulation in Manufacturing and Defense Systems Acquisition*, National Academies Press, Washington, DC, 2002.

Sage, A. P. (1992). *Systems Engineering*. New York: Wiley.

Sage, A. P. (1995). *Systems Management for Information Technology and Software Engineering*. New York: Wiley.

Sage, A. P., and Cuppan, C. D. (2001). On the Systems Engineering and Management of Systems of Systems and Federations of Systems. *Information, Knowledge, and Systems Management*, *2*(4), 325–345.

Sage, A. P., and Lynch, C. L. (1998, November). Systems Integration and Architecting: An Overview of Principles, Practices, and Perspectives. *Systems Engineering*, *1*(3), 176–227.

Sage, A. P., and Rouse, W. B. (Eds.). (1999). *Handbook of Systems Engineering and Management*. New York: Wiley.

Sanders, P. (1997, September/October). Simulation Based Acquisition: An Effective, Affordable Mechanism for Fielding Complex Technologies. *Program Manager*, pp. 72–77. Available: www.dsmc.dsm.mil/pubs/pdf/pmpdf97/sanders.pdf.

Shapiro, C., and Varian, H. R. (1999). *Information Rules: A Strategic Guide to the Networked Economy*, Boston, MA: Harvard Business School Press.

U.S. Army. (2000, September). *Planning Guidelines for Simulation and Modeling for Acquisition, Requirements, and Training*. Available: www.amso.army.mil/smart/documents/guidelines/index.htm.

U.S. Department of Defense. (2000a, October 23). *The Defense Acquisition System* (Directive 5000.1) and *Operation of the Defense Acquisition System* (Instruction 5000.2). Available: ric.crane. navy.mil.

U.S. Department of Defense. (2000b). Acquisition Reform website: www.acq.osd.mil/ar.

U.S. Navy. (2001, February). *Modeling and Simulation Verification, Validation, and Accreditation Handbook*. Available: navmso.hq.navy.mil.