5. TO-BE ARCHITECTURE

5.1 INTRODUCTION

This section describes the operational, systems, and technical architecture views that provide a target on which Government and Industry can focus their sights to achieve the stated SBA vision and goals in the future. These architectures, prepared using the process described in Section 1, are based on a snapshot in time and should be evolved based on cultural, process, and environmental changes; state-of-the-art technologies; and a variety of other factors that may impact the Acquisition Management System. Although these SBA architectures need to be adaptable and flexible, just like our DoD acquisition and warfighting systems, there needs to be a degree of stability in the structure so that stakeholders can build toward a common vision and goals to preclude duplication of effort and avoid making processes even more complicated than they are today.

Conceptually, operational architectures would include information flows indicating data types and to whom they should be passed. However, depicting information flows to satisfy Service, program, and even Program Manager unique processes is varied and complicated and beyond the scope of this task. Consequently, the “to-be” operational architecture presented represents an aggregation to present a notional operational architecture for DoD acquisition. The Task Force recognizes the requirement to eventually identify the detailed flows. However, any attempt to derive overarching information flows under the possible solution set was not considered to be within the scope of the Terms of Reference (TOR). However, the Task Force acknowledges their value and believes them to be so specific to individual established programs and environments that the notional to-be operational architecture addresses them through the concept of collaborative environments. Within the context of SBA, a collaborative environment (CE) is an enduring collection of subject matter experts (SMEs) supported by interoperable tools and data bases, authoritative information resources, and product/process models that are focused on a common domain or set of problems.

An example of a collaborative environment that is referred to later in this section encompasses the strike warfare mission area. Participants would include organizations involved in making sound acquisition decisions such as those responsible for requirements generation and oversight, program management, cost analysis, training, maintenance, logistics, manufacturing, and command and control (C^2) systems. It is envisioned that such an environment would contribute to the total life cycle management of a system through analyses of alternatives, requirements refinement, and system design and trade-off studies, to name a few. This particular collaborative environment would allow different programs involved in strike warfare to jointly use the same resources, promoting reuse and interoperability.

The to-be architectures defined in this section are an attempt to add additional structure to SBA based on the Task Force's review of architectures and the experiences/recommendations from a broad community that has a stake in the successful implementation of SBA. Incorporating these to-be architectures, the underlying elements of these architectures, and the recommendations provided in Section 7 will accelerate progress toward achieving the SBA vision and goals. With the exception of three, lower-level process issues (items 5–7 under Process-Related Shortfalls in Section 3.3), the to-be architecture views provided in this section directly address the

shortfalls cited in Section 3. The to-be architectures would need further elaboration (beyond the scope of this task) to show direct linkages to these shortfalls, but these items were still included in the road map because of their significance. As with all complex, potential high-payoff endeavors, there are a series of challenges that must be met to achieve the vision and goals of SBA. There are also some barriers that will inhibit progress in certain areas unless they are overcome. The challenges and barriers to implementing the SBA architectures are also listed in this section.

5.2 TO-BE OPERATIONAL ARCHITECTURE

Description

When Figure 3-2 is examined (see Section 3), a need does not appear to exist for making radical changes in the basic organizational elements, activities, military functions, and support systems to implement SBA. However, even from this global view, an opportunity appears to exist for stakeholders such as Industry to become more involved in activities outside development and production areas on which they have concentrated in the past. The to-be collaborative environment shown in Figure 5-1 emphasizes the concept of bringing Government, Industry, and Academia closer together through inter-functional rather than intra-functional IPTs.

Having sets of integrated/interoperable tools that would allow M&S tools and data to be shared among different functional areas would improve coordination between functional area experts. Integrated M&S tools would also ease the chore PMOs have with maintaining configuration control over a core suite of tools used within the program and in performing verification, validation, and accreditation (VV&A) of these tools. It would also increase the opportunities to reuse M&S tools and data because needs for functional areas that receive greater emphasis later in the program would be able to influence the M&S program early on.

Having integrated data bases would also relieve a tremendous load on the program IPTs, who have to sift through large volumes of information and hold extra face-to-face meetings to find data "nuggets" and to plan M&S activities that are applicable to their own functional areas. It is expected that this reduction in the workload for intra-function IPTs would free up people to exchange ideas with other IPTs within the program and with IPT members of other programs that have common M&S needs.

The collaborative environment shown in Figure 5-1 would help break many of the stovepipes that exist not only within a program but also outside a program. Current suppliers such as the Defense Intelligence Agency (DIA) are tasked to provide large quantities of specialized threat data to a myriad of M&S developers and users. Providing these data undoubtedly creates an extra burden on staff who have to contend with personnel downsizing at the same time. While fewer major programs today demand attention from organizations such as DIA, an inordinate number of tools are still being used and new ones that require vast amounts of data are being built at a rapid rate.

Having integrated teams, tools, and data bases for PMOs would be a major step toward the SBA vision. Another important way to recognize the full potential of SBA is to give PMOs an opportunity to use investments made by programs outside the respective PMO's sphere of authority and responsibility by allowing the PMOs access to the programs' capabilities without

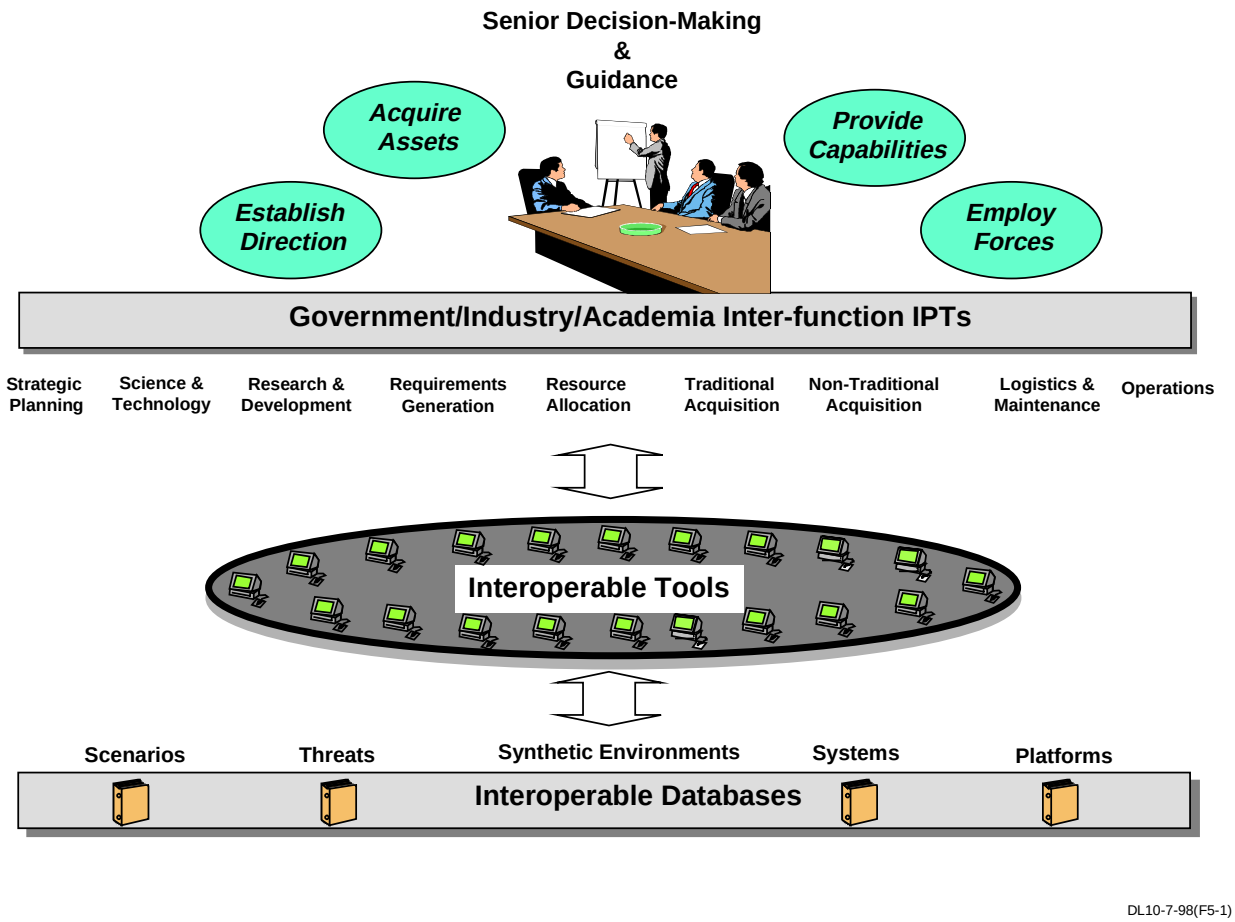


Figure 5-1. To-be collaborative environment.

having to pay for them. This opportunity would not be limited to M&S, but would also include supporting data; programmatic, operational, and technical expertise of various program staffs; and access to other customers of the organizations that are responsible for developing, maintaining, operating, and reporting the results of M&S activities that may be relevant to other programs.

Establishing collaborative environments, as shown in Figure 5-2, is an important way to greatly increase sharing of resources among acquisition stakeholders. The structure shown is an approach to defining levels, which can evolve naturally as the result of explicit efforts. The first two levels of collaborative environment, strategic and operational, align with the levels of conflict described in Joint doctrine publications. The next level is the tactical mission level of conflict, where multiple systems are brought together to perform specific missions or functions that each of the Services must be prepared to perform in fulfillment of their Title X roles. Mission-level collaborative environments are particularly important because they would support up-front mission area assessment (MAA) and analysis that helps justify new programs and provides insight into the utility of new systems and concepts of operation (CONOPS) throughout the product life cycle. The ability of a new weapon system to perform the mission as part of the integrated System of Systems (SoS) is the bottom line for acquisition, training, and operational employment, which further emphasizes the need for mission area collaborative environments.

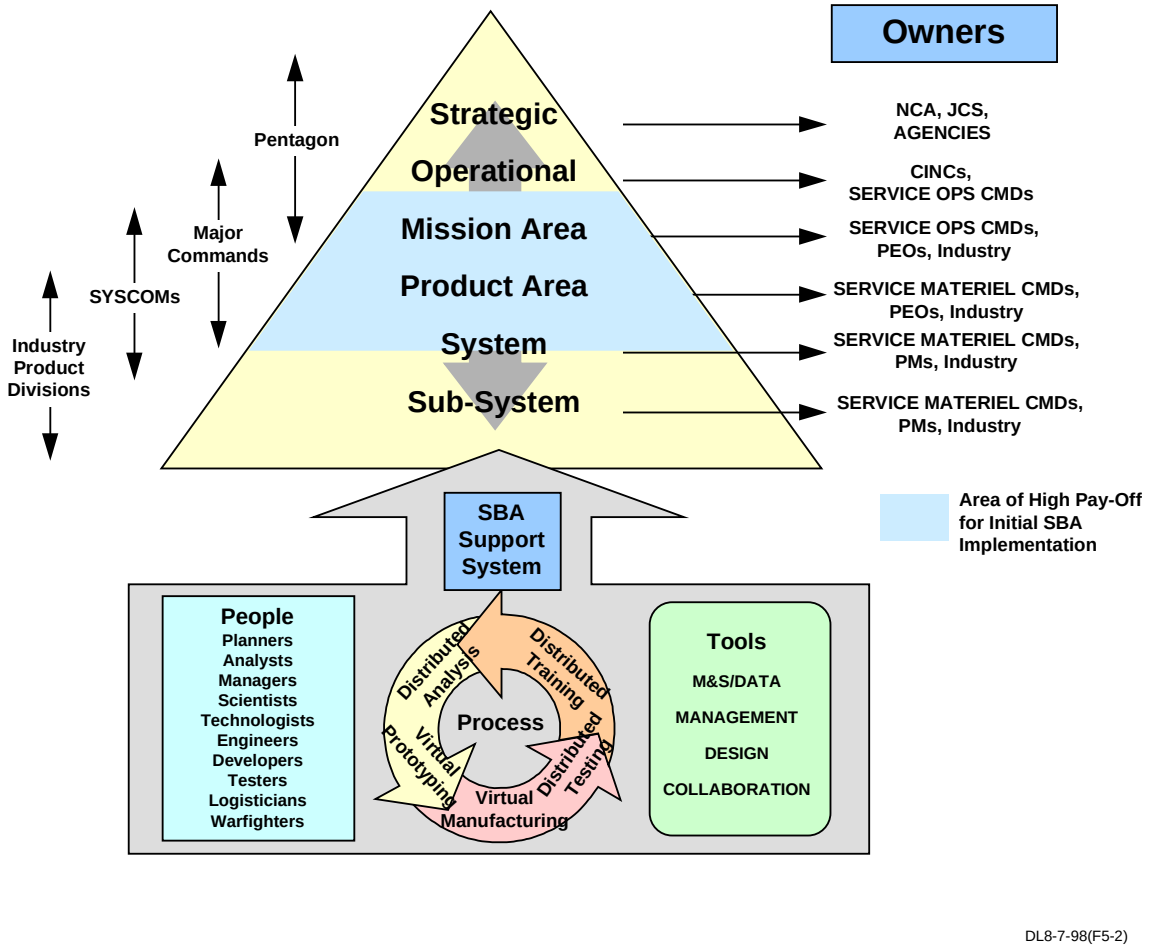


Figure 5-2. Proposed levels of to-be collaborative environments.

The next three levels of collaborative environment, product area, system, and sub-system, are natural alignments from an acquisition perspective, just as the first three levels are natural alignments from an operational perspective. Service materiel commands are responsible for overseeing the development and fielding of systems. Product areas are usually established by general categories of platform (e.g., aircraft, missiles, track vehicles, surface ships that support littoral warfare) within the different Services in order to integrate similar programs. Many examples exist of partial implementation of these collaborative environments at the product area and systems level; examples also exist of limited implementations in the operations-related areas. However, there are inconsistencies in how these environments are defined along with significant variations in the M&S capabilities within these environments that inhibit interoperability and the sharing of resources across programs and program phases.

One of the key ingredients that would allow these environments to promote more effective collaboration is a robust SBA support system. Such a support system would consist of the right people, processes, and tools to meet the special needs of a given collaborative environment and also be capable of supporting collaborative environments at other levels. PMOs could depend on these environments to have long-term staying power, i.e., these environments would be available throughout the life cycle of the program. PMOs could reduce the size of IPTs responsible for

providing M&S support by relying more heavily on experts who are permanently assigned to support these collaborative environments. Although a need exits to educate Government and Industry on M&S and SBA, in general, it is not necessary (or realistic) to have M&S experts on all the IPTs that will rely on M&S to meet their particular needs. Many of the components that are key to these support systems exist today under the auspices of the Service laboratories and centers. Every effort should be made to use their efforts and resources to ensure robustness in the SBA support system.

Having collaborative environments that are well defined, recognized, institutionalized, and adequately funded would reduce duplication and the dilution of investments in SBA. Defining ownership is a critical step in the process of making SBA happen. The more closely ownership for these environments is aligned with the organizations that have "real-world" responsibilities (in contrast to M&S responsibilities), the better. This tenet is in line with the adage "train the way we fight," only for SBA, it would be "model and simulate the way we plan, acquire, train, and fight." The specific definition and implementation of collaborate environments within the Military Departments should be left up to each Department. Considerations such as organization/ PMO alignment, mission/functional responsibilities, funding elements, and M&S capabilities would have a major influence on collaborative environment structures both within and between OSD, JCS, Services, and Industry. The long-term development of these CEs will be influenced by the results of shorter-term experiments implemented by individual programs to implement SBA concepts in support of these programs.

Another key requirement of the SBA support system is that it must accommodate distributed activities. Stakeholders need to gain access to information and infrastructure much more quickly than they can today if SBA is going to have high payoff. This does not mean that all activities must be distributed—only the right ones at the right time during the life cycle. For instance, a great benefit is still derived from bringing the community together on a common range to accomplish a joint test or training exercise. There will always be a place for direct feedback and interaction.

However, declining budgets and personnel levels and high operations tempos continue to increase stress on staffs and operational/programmatic risks. Finding new ways of doing business is the only way to stop the erosion of capability. Distributed simulation is one way to save wear and tear on people and real-world equipment and exploit new ways to minimize cost and risk. Allowing stakeholders to accomplish their missions from their permanent duty locations can be one of those force multipliers that actually allows the DoD to reduce temporary duty (TDY) assignments. Having distributed simulation capability at home stations can give developers, testers, operational crews, maintainers, and logisticians access to realistic environments that cannot be created in open-air tests or exercises because of constraints such as security, safety, scheduling, and funding. Providing all stakeholders ready access to product models and data through distributed networks and repositories can significantly increase efficiency as well.

Figure 5-3 provides a more practical example of the proposed to-be collaborative environment starting from the strategic level and going down to the weapons system level under the strike warfare mission area. At the strategic level, high-level military objectives and tasks in support of national strategy are developed by senior authorities like the National Command Authority (NCA), Chairman of the Joint Chiefs of Staff (CJCS), CINC, and Joint Force

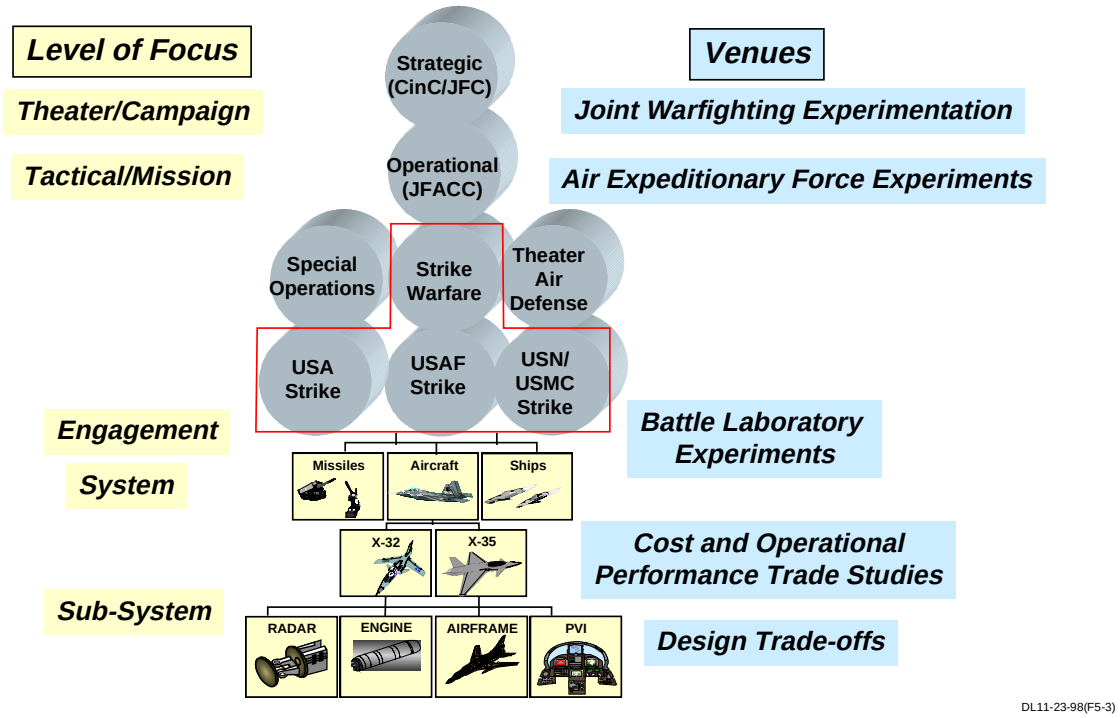


Figure 5-3. Collaborative environment for strike warfare.

Commander (JFC).[25] The operational level links tactical employment to strategic objectives and is focused on employing military forces to attain objectives through design, organization, integration, and conduct of strategies, campaigns, major operations, and battles.[25] In Figure 5-3, the Joint Force Air Component Commander (JFACC) is used as an example of a Joint collaborative environment that would focus on employing the operational art of war to support the mission and objectives laid out by the CINC/JFC. Numbered armies, fleets, and air forces are specific examples of organizations that concentrate on the operational level within Service boundaries.

For the purposes of this road map, the term "mission area" is used to define one of the primary levels of the collaborative environment. The DSMC glossary defines mission area as "a segment of the defense mission established by SECDEF. Each component has mission areas (e.g., Navy-anti-surface warfare, Army-ground warfare, etc.) for which it must equip its forces."[26] The term mission area appears to be roughly equivalent to functional areas, which are defined in Joint Publication O-2 as "specific responsibilities assigned by the SECDEF to enable the Services to fulfill their legally established roles."[17] Examples of Service functional areas are as follows: Army—Fire Support, Air Force—Interdiction, Navy—Surface Warfare, and Marine Corps—Close Air Support. Some evolution continues within the DoD on terminology and definitions with respect to mission and functional areas as doctrine documents are updated. However, the level in which both terms seem to apply is at the tactical level of warfare where combat units are arranged and maneuvered in relation to each other and to the enemy "to fight and win engagements and battles."[26]

Above the mission level, the operational level commander is responsible for the integration of forces provided by the Services to accomplish multiple missions. When a simulation test or exercise needs to involve other Services, the natural integration points would probably be at the operational, mission area, or product area, depending on the situation. The Services should be

provided with flexibility to integrate with collaborative environments at various levels based on their own preferences and capabilities. However, the closer this connection mirrors the level at which systems are integrated for real-world operations and product development and integration, the better.

Below the mission area, Figure 5-3 shows the next level of collaborative environment as the product area, with product areas in this case consisting of missiles, aircraft, and ships. The office of primary responsibility (OPR) for this collaborative environment would focus on integration of multiple platforms to support the mission areas. As an example, if the JSF were the "system under test," contractor collaborative environments could be linked to the higher echelons as part of the JSF collaborative environment. At the lowest level in the chain, sub-components of the JSF, such as radar, engines, airframe, and pilot–vehicle interface, would be part of the JSF collaborative environment.

It would be appropriate to follow the same tiered, collaborative environment approach for support areas such as threats and synthetic environments. In the case of threats, collaborative environments should span the entire set of tiers from strategic to sub-system level. For synthetic environments such as atmosphere, ocean, and terrain, it would not be necessary to have collaborative environments above the product area level because the mission/operational context does not alter the physical environment of synthetic representations of it. As an example, in the case of terrain, the product area would be responsible for assembly and dissemination of imagery topography, and geodesy data/models to the various operations and systems-related collaborative environments. There would be no need for a mission area, operational, or, strategic-level terrain collaborative environment, but there would certainly be a need to have threat systems and CONOPS represented at these various levels.

From a collaborative environment concept of operations point of view, it is envisioned that customers tapping into environments at each of the levels described would have ready access to constructive, virtual, and live simulation capability and related data. These collaborative environments essentially would be distributed repositories of authoritative representations of individual systems that fall within operational or product domains. At the system level, the PMO would be responsible for the digital representation of its system capability.

Moving up to the product area level, this collaborative environment would provide the repository for maintaining various tools and infrastructures applicable across multiple platforms within their area that are optimized for technology/product development, developmental testing, and integration. At the mission level, for example, the collaborative environment OPR would provide validated, mission-level constructive models and for assembling virtual and live assets that are required to meet mission-level, operational testing and exercise objectives. These same themes would apply to the operational level, only collaborative environments at this level would concentrate at the theater or campaign level and provide linkage to the strategic level when necessary.

Figure 5-3 also shows some existing venues in which acquisition stakeholders have potential opportunities to look at new technologies; systems; CONOPS; and tactics, techniques, and procedures (TTP) within the context of SoS at the various levels. Many of these venues provide an excellent environment for looking at technologies, systems, and concepts that can be fielded in the near term. An open issue still exists on how well these experiments and exercises will adequately support programs like JSF that look at C4ISR architectures, technologies, and employment concepts for a 2010 SoS environment.

Again, using strike warfare as an example, Figure 5-4 provides an expanded view of what the Task Force believes to be the areas of high payoff for initial SBA implementation of collaborative environments—Mission Area and Product Area. The primary reason that these two levels stand out is that they have the greatest potential to provide an integrating function across programs and phases, which is a major goal of SBA. They are also well aligned with the ways we conduct integrated operations and the way we manage acquisition of systems, i.e., the Mission Area by accomplishing integrated missions in support of the operational-level commander and the Product Area by managing multiple products to help realize economies of scale. The following lists provide a short summary of key responsibilities for Mission Area and Product Area collaborative environments. Again, the specifics of how these collaborative environments are defined and how these responsibilities are assigned should be left up to the individual Services.

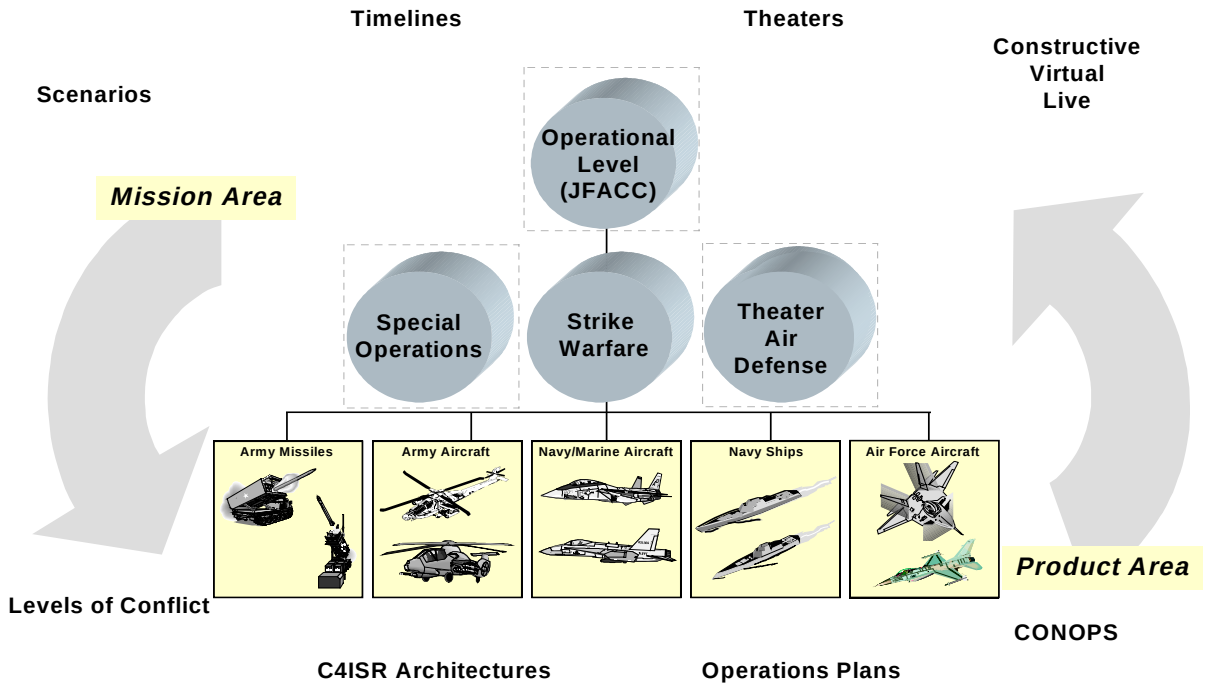


Figure 5-4. Collaborative environment for mission and product areas.

Mission Area Collaborative Environment Responsibilities

1. Integrating multiple systems within an operational context
2. Communicating operational requirements within a mission area
3. Assembling a command and control environment in which these various systems will interoperate
4. Providing operational orientation to the product development community
5. Assembling resources that facilitate CONOPS, TTP, and operations plan development; operational T&E; and operational training

6. Supporting MAA, analysis, and Joint Warfighting Experimentation (JWE) to determine mission needs and the potential utility of new concepts, technologies, and systems to improve mission performance and effectiveness

Product Area Collaborative Environment Responsibilities

1. Integrating multiple programs and systems
2. Supporting management of multiple programs
3. Assembling resources that facilitate research, development, test, and evaluation (RDT&E) (i.e., developmental T&E)
4. Providing technical orientation to the operational community
5. Determining the technical feasibility of a new technology or system
6. Preparing a new technology or system for insertion into the operational assessment environment
7. Hardening/ruggedizing a new system prior to fielding
8. Providing a focusing mechanism to Government and Industry for science and technology and research and development initiatives

Within product areas, collaborative environment OPRs would be responsible for helping acquisition managers look at the technical aspects of their systems, such as technical feasibility, system or technical architecture alternatives, and integration with other existing or developing systems. Once a concept has matured through this environment through iterative improvement cycles between the product area and system-level collaborative environments, it could be introduced into the mission and higher level environments, when appropriate, to perform operational utility and feasibility assessments. Results from the operational assessment would be fed back into the product and systems area environments, which would lead to further refinements of the technology, system, or concept and then be re-inserted into the mission, operational, and/or strategic environments for final checkout prior to final packaging and fielding. For a major weapon system acquisition program, the product would go through multiple iterations between the product and mission/operational/strategic environments. In the case of mature and short life-cycle technologies, such as commercial-off-the-shelf (COTs) information technologies, this spiral process should occur very quickly so that technologies with short "half-lives" would get into the hands of warfighters before the technologies become obsolete.

The spiral, iterative development process referred to in the previous paragraph is currently being implemented to varying degrees within the Service boundaries for the most part. The to-be operational architecture for SBA will go well beyond the current implementation of spiral development. The SBA operational architecture of the future must mature to a point where mission needs, requirements, and systems are developed and tested in a Joint operational context though day-to-day interactions between collaborative environments of each of the Services. A considerable amount of time and effort will need to be devoted to achieving this to-be architecture that provides flexibility in all the dimensions noted in Figure 5-4 (e.g., interaction between collaborative environments at various levels, for different theaters, time frames, and scenarios).

From the collaborative environment standpoint, it is also important that the right people be brought together at the right time to facilitate IPPD. Figure 5-5 shows the primary types of people that need to be available to make the collaborative environments work. One goal would be to

have a substantial number of these individuals in permanent billets to provide continuity. While IPT members would be a healthy percentage of the groups that interact in the collaborative environment, they should not be depended upon to operate and maintain much of the infrastructure because this would detract from their primary mission of providing direct support to PMOs.

The conceptual model shown in Figure 5-6 provides the last view of the proposed to-be operational architecture—the to-be SBA process. A number of high-level process models have been developed for SBA from various points of view. The DTSE&E STEP model and SBA model described in the draft Functional Description Document (FDD) produced by the Acquisition Division of the M&S Industry Steering Group (ISG) are two of the more comprehensive views of SBA.[2,27] Several programs, such as DD-21 and JSF, have developed useful descriptions of SBA from program perspectives. Both higher-level and lower-level views were considered in the development of the to-be SBA process model illustrated in Figure 5-6.

One of the key requirements of the to-be SBA process is that it must accommodate sharing of collaborative environments across functional and program boundaries. For example, individuals who are responsible for performing requirements analysis within a program office should be able to benefit directly from the resources that were invested earlier in the life cycle for MAA, analysis of alternatives (AOA), and Joint Warfighting Capabilities Assessment (JWCA).

In the same vein, those individuals conducting MAAs, AOAs, and JWCAs should be able to directly benefit from program office investments in more accurate product models and data that were developed through an iterative process during performance of cost and operational performance trades (COPT), virtual prototyping, and developmental/operational test and evaluation (D/OT&E).

This same concept applies to leveraging investments in other programs like Advanced Technology Demonstrations (ATDs), ACTDs, Joint Warfighting Experimentation (JWE), and Joint Tests and Evaluation (JT&E). (The purpose of the OSD JT&E program is to assess the interoperability of Service systems in Joint operations, evaluate Joint technical and operational concepts, and validate testing methodologies that have multi-Service application.) Decision-makers, warfighters, and maintainers should also be able to take advantage of investments made at earlier phases of the DoD enterprise to support unit training, CINC exercises (CINCEXs), Follow-on Operational Test and Evaluation (FOT&E), combat mission preparation, and force employment. For example, virtual simulators that are developed during Program Definition and Risk Reduction (PDRR) could be evolved through iterative development and testing and become virtual test devices for Engineering and Manufacturing Development (EMD). Eventually, these virtual test articles could be evolved into distributed training devices for PF/DOS. As part of the ongoing, iterative process, quantitative and qualitative data, along with lessons learned, would be collected from decision-makers, warfighters, and maintainers, based on field experiences, and provided as feedback to further enhance collaborative environments and, ultimately, the quality of decisions being made throughout the DoD enterprise.

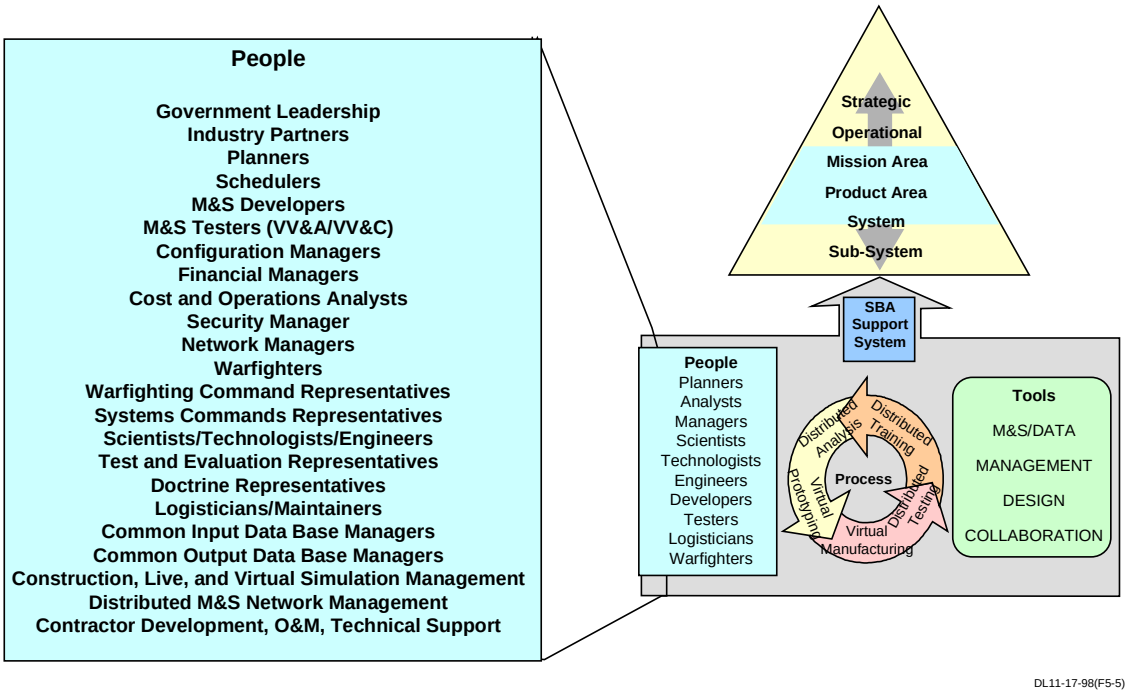


Figure 5-5. Collaborative environment personnel.

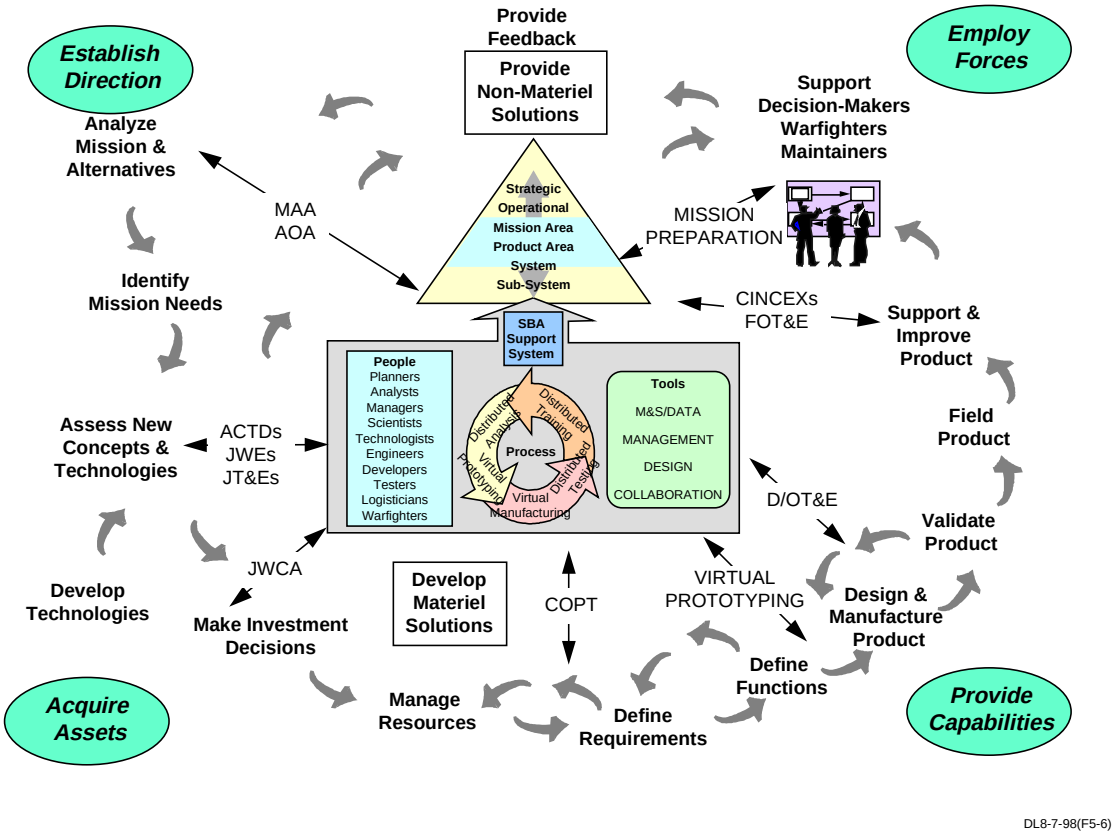


Figure 5-6. To-be SBA process model.

In essence, the SBA support system, which consists of people, processes, and tools, would be the "critical engine" for collaborative environments that enable IPPD across programs and phases. Having collaborative environments that support processes within multiple functional areas will improve chances for cooperation communities that have traditionally operated within the confines of their own functional areas. In particular, this process should give Industry additional windows of opportunity to become more involved in activities such as strategic planning, identification of mission needs, and training, areas in which Government personnel have been involved almost exclusively in the past, other than providing technical and administrative support. Insights gained by Industry through participation in a broader range of activities within the DoD enterprise will also stimulate innovation in the area of developing SoS technologies and concepts for SBA that can be extended to other activities of the DoD enterprise that are connected with acquisition management.

When the current acquisition process model in Figure 3-5 (see Section 3.3) is compared with the to-be SBA process model in Figure 5-6, the most significant difference is found in the notion that customers need to tap into more than just an M&S tool environment; they need to tap into a collaborative environment that consists of the right blend of people, processes, and tools. Another difference is that the to-be model in Figure 5-6 tries to acknowledge the utility of tying other communities, such as strategic planners and assessors and nontraditional program managers, into the traditional acquisition management to allow for mutual leverage. Figure 5-6 also emphasizes the idea of a spiral or iterative process rather than a waterfall process that leads to the development and fielding of a new weapon or information system. One of the common themes reflected in both the as-is and to-be process views is that M&S is a key tool in reducing operational and programmatic risk.

The to-be process model also emphasizes the role of technology in providing a "push" toward new capabilities in addition to the operational needs "pull." Effectively merging the strategic planning and requirements communities with Industry, DARPA, the National and Service laboratories, Federally Funded Research and Development Centers (FFRDCs), and Academia at the front end of the process is a requirement of the acquisition process that can be facilitated through SBA. The Government needs to be able to look at new technologies that are developed through Independent Research and Development (IR&D), Small Business Innovative Research (SBIR) programs, and Dual Use Application Programs (DUAPs) as technology maturation and risk reduction measures at the front end of the life cycle. Operational context would be added to these technologies through ATDs, ACTDs, JWEs, and JT&Es. Close ties between Government and Industry early on should help eliminate false starts and unexpected stumbling blocks for major programs. Although these concepts are now being exercised, a tremendous opportunity exists to bring Government and Industry even closer together through institutionalizing SBA.

Issues

Considerable effort is under way in the community to overcome many of the shortfalls cited in this document and various other source materials. This section provides a summary of some of the challenges that lie ahead with respect to implementing the to-be architecture proposed in this road map. Several barriers also stand in the way of achieving the to-be operational architecture. Environment-related (i.e., technical) and process-related challenges and barriers are listed in the following paragraphs. Cultural, systems, and technical challenges and environment-related

barriers that stand in the way of achieving the to-be operational architecture are discussed in other parts of this document.

Challenges

Challenges can be divided into two categories: environment-related and process-related.

Environment-Related Challenges

1. Each of the Services is organized along various lines based on roles, mission, functions, business processes, relationships with Industry, and a host of other factors to meet their individual needs. Building integrated, collaborative environments that allow for efficient interplay, both vertically and horizontally across various Service boundaries, will take a significant amount of time and effort. Some experimentation will be necessary to determine the best organizational relationships within and external to each Service and the right balance of responsibilities between permanent party personnel and IPT members.
2. Potentially, collaborative environments could become unwieldy, i.e., so large and so heavily tasked that they will not have the necessary agility to respond to customers to meet their timelines and objectives. Considerable attention will need to be devoted to ensuring that collaborative environments do not have single points of failure. In the same way that real-world systems architectures are being built for redundancy, the SBA operational architecture must be built to allow for "graceful degradation" if components fail, rather than a catastrophic failure. The complexity of an integrated SBA architecture will require on-going research to address the issues of scalability and vulnerability as these environments grow.
3. The process of preparing and executing distributed tests and exercises between geographically distributed sites has historically posed an ominous management challenge, primarily because of the complexity of getting M&S to interoperate effectively.
4. It will be an on-going challenge to develop and maintain M&S capabilities that will support a wide range of customers who have different needs. For example, providing a simulation that satisfies the needs of the testing and training communities will require increased coordination. Another related challenge is the successful execution of systems test activities during training exercises without compromising the objectives of each discipline. Government and Industry must avoid the pitfall of sub-optimization of these collaborative environments where they become “jacks-of-all-trades and masters of none.”
5. It will be difficult to compose collaborative environments across the Services that are relatively consistent in fidelity at each level in the hierarchy, particularly when a more aggregated environment attempts to import tools developed originally for more detailed applications at the levels. Significant variations in level of detail or accuracy of M&S and data shared among collaborative environments can cause unrealistic results, to the detriment of one or more of the collaborators. It will also be difficult to effectively import the M&S and data from the lower levels up to higher levels.

Process-Related Challenges

1. It will be difficult to build standardized, yet flexible, process models for the collaborative environments that meet the needs of multiple programs given the current amount of

variability between programs (schedules, priorities, funding, etc.). Significant effort will be required to assimilate and characterize process models that are used by various programs to determine how programs could transition to more integrated processes without having a significant impact on efficiency. Detailed process models will also need to be developed to determine the potential ROI for SBA by investigating the resources required to accomplish processes in the current state and by projecting the impacts of changing to a new way of doing business.

2. Once collaborative environments become operational, it is expected that there will be large numbers of customers. It will be a significant challenge to make M&S capabilities available at the right times to meet the particular needs of all these customers. Undoubtedly, conflicts in resource priorities will need to be resolved between the stakeholders.
3. Financial management will pose a major challenge to OPRs and users of these collaborative environments. There will be different expectations about the amount of funds that will be provided by one party versus another for access to facilities and services. Customers will also bring different levels of funding and colors of money to the table and have specialized needs based on their M&S needs and capabilities of their own staffs. Developing, coordinating, and implementing integrated investment plans across functional areas, phases, and multiple programs will pose a formidable challenge.
4. Transition from the current SBA process to the to-be process without disrupting on-going programs will require careful planning.
5. The process by which people are selected and rotated through leadership positions in these collaborative environments needs to be developed to ensure that the right talent is brought to bear in these higher-risk/higher-payoff areas.
6. Matrix management associated with IPTs can also pose a challenge, given that members have significant commitments to their formal organizations, which they have to deal with along with their IPT commitments, all of which can be quite dynamic.

Barriers

Although there are environment-related barriers, no process-related barriers were identified that would prevent successful implementation of SBA.

Environment-Related Barriers

1. Programs do not operate at the same levels of security classification. This imposes restrictions on the degree to which programs may interact and share resources within and between collaborative environments.
2. Proprietary information and property restrictions inhibit sharing of resources and collaboration.

5.3 TO-BE SYSTEMS ARCHITECTURE

This section describes an overarching architectural design for the classes of M&S tools, resources, and other components needed to enable the “to-be” operational view of the systems acquisition process.

Purpose

Section 5.2 in this document has described a future vision for how M&S may be applied more effectively and efficiently in the systems acquisition process. This vision has thus far focused on the operational view of the SBA architecture, for which the collaborative environment was identified as the fundamental construct. The primary purpose of this section is to describe the hardware and software components needed to enable this operational view, along with a proposed structure for how these components logically interrelate.

There are three main reasons for specifying an architecture for supporting hardware and software components. First, it is important to explicitly define all the necessary components of an SBA systems architecture and their interrelationships so as to lay out a blueprint for what is needed and to highlight any systems-level technical challenges that must be addressed to reach the desired operational state. Second, the relationships between systems components defined by the architecture are helpful for identifying opportunities for interface standards. Third, opportunities for cross-program reuse and interoperability of systems components may be facilitated by a common architectural structure, which can only be adopted through its explicit specification.

The systems architecture described in this section leverages heavily off a vast amount of previously documented knowledge and experience in a wide variety of related subjects. For instance, existing architectural frameworks such as the DoD Technical Reference Model (TRM)[27a] and the C4ISR Architecture Framework[14] were useful in defining a high-level construct for the SBA systems architecture. Many documents, technical papers, and reports from technical conferences, workshops, and symposia were instrumental in identifying the required components and overall functionality of an SBA Systems Architecture, although few actually defined the component interrelationships needed for an architecture specification. Finally, actual programs (e.g., Distributed Knowledge Environment (DKE), Joint Strike Fighter (JSF), etc.) that are experimenting with systems-level SBA-related concepts were consulted in order to appropriately leverage their "lessons learned." (Formerly known as the Simulation-Based Design (SBD) program, DKE is developing an architecture that supports virtual prototypes in a synthetic environment to enable Integrated Product and Process Development (IPPD).) Thus, the goal throughout the development of this systems architecture specification has not been to reinvent an entirely new systems concept, but rather to integrate, repackage, and (in some cases) reengineer previously developed concepts into a coherent, cohesive design structure that fully supports the SBA operational architecture and can be realized (in varying degrees of sophistication) within a relatively short time frame. The specific sources used to develop this architecture are provided in Section 8, References and Bibliography, of this report.

Requirements

As stated earlier, several past studies have attempted to identify a basic set of needs or requirements for a SBA systems architecture. Each of these studies has (at least indirectly) stated its requirements in terms of a perceived vision for an operational "end state." Although the details of these end states differed from report to report, there was also broad recognition of many fundamental operating principles for modern SBA processes and procedures that have implications at the system level. Examples of these types of principles include the need for software reuse and interoperability both within and across acquisition programs, the need for data standards, and the need for new tools to support the emerging SBA operational architecture. The

importance of these basic concepts has been validated by the SBA Task Force via the Quality Function Deployment (QFD) process.

The following list of technical requirements is a summary (and aggregation) of the system-level needs that have been identified and derived from past SBA-related studies and of new systems requirements that have been identified during development of the SBA operational architecture. Note that these needs have not been driven by what modern hardware and software technology can support; rather, the intent of the systems architecture is to drive the development of new supporting technologies, when necessary, based on stated operational needs. For instance, although supporting technologies to support the processing of human language have advanced rapidly in recent years, the mere existence of such a technology is not a sufficient reason for including such components into a systems architecture designed for SBA applications. On the other hand, the relative immaturity of supporting technologies to accurately replicate human reasoning and decision-making is not a sufficient reason for omitting such "intelligent systems" components from the architecture given that there is a recognized need for them. The types of technology advancements that must take place to fully implement this architecture are highlighted in Section 7, Implementation Recommendations, of this report.

- The SBA systems architecture must facilitate effective use of M&S within an enterprise, and generally enable the systems acquisition process via ready, dependable access to supporting tools, resources, and infrastructure.
- The SBA systems architecture must facilitate interoperability between, and reuse of, models, simulations, data, and other infrastructure components (networks, facilities, etc.) across different functional disciplines and programs and between Government and Industry.
- The SBA systems architecture must conform to all applicable DoD and/or commercial standards.
- The SBA systems architecture must make maximum use of existing Government/commercial off-the-shelf (GOTS/COTS) products and Government R&D programs to minimize new tool and infrastructure development.
- The SBA systems architecture must be open, scaleable, and flexible, allowing many different implementation approaches.
- The SBA systems architecture must operate effectively in distributed, heterogeneous computing environments and must include secure, high-bandwidth communications to effectively use these resources.
- The SBA systems architecture must enable access to and use of authoritative, verified and validated models and simulations (and associated data) as necessary to support the systems acquisition process.
- The SBA systems architecture must include reusable representations of the natural environment and other "general-purpose" data bases in standardized formats.
- The SBA systems architecture must include ready access to authoritative functional battlespace descriptions to facilitate system model development.

- The SBA systems architecture must include a DoD- and Industry-wide repository of acquisition support resources, with “intelligent” mechanisms to guide users to desired resources and appropriate controls to protect classified data and proprietary data rights for Industry.
- The SBA systems architecture must facilitate collaboration between members of geographically distributed Integrated Product Teams (IPTs), including mechanisms for distributed IPTs to operate from a common, standards-based, internally consistent product description (data, behavior, and associated development processes).

Description

Introduction

The breadth of the technical needs specified above highlights the extremely broad scope of the SBA systems architecture. This architecture must support a wide variety of diverse user communities and must include a comprehensive set of tools and supporting infrastructure to enable Integrated Product and Process Development (IPPD) within and across those communities. At first, the prospect of defining an architecture of this magnitude seems daunting. However, it is important to realize that most of the individual components of an SBA systems architecture can be readily identified from the classes of tools, resources, and infrastructure used in present-day acquisition programs, and other necessary components can be easily projected based on evolving SBA operational concepts. Thus, the primary thrust in specifying a systems architecture design for SBA concerns two main issues: (1) how the various components need to interrelate to maximize the efficiency of development and/or use, and (2) identifying and addressing the technology deficiencies that impede a complete instantiation of the architecture.

The strategy for maximizing efficiency within an architecture depends on the perspective from which efficiency is viewed. That is, architectural efficiency can refer to the ease by which an architecture is implemented, or to the overall performance of implemented systems that conform to the architecture, or to the speed with which an implemented system can be customized for new applications. To choose a perspective for the SBA systems architecture, one only need to consider the primary “showstoppers” encountered when trying to apply IPPD in today’s acquisition environment. Generally, there has been little argument in the technical literature regarding the potential benefits of closely integrated cross-functional teams working collaboratively to develop and optimize a system concept. However, the practical benefits promised by IPPD depend heavily on a free, unrestricted flow of digital information across the various types and classes of automated tools that support the acquisition process. Besides the obvious need for complete, robust sets of these tools to exist, a serious deficiency that exists in today’s acquisition environment is that the automated tools and resources that support the acquisition process (including models and simulations) tend to be stovepiped by program and functional discipline. These tools and resources also generally possess incompatible data formats, which makes automated data exchange impossible. Thus, mechanisms for facilitating data transfer and reuse must be paramount in the architecture, and represent the primary drivers for defining the perspective from which architectural efficiency and overall “goodness” can be determined.

This need for seamless data transfer and reuse of M&S is particularly important in virtual prototyping and virtual manufacturing environments. A virtual prototype is a computer software module that models the structure and behavior of a product, in various stages of development, within simulated environments for the purpose of design optimization. Virtual manufacturing refers to the use of manufacturing simulations to optimize the design of a product for specific manufacturing goals and/or to analyze and optimize production processes and factory operations. In both cases, the cost-effectiveness and eventual success of these design and development paradigms are highly dependent on the speed and reliability with which the relevant characteristics of a particular virtual prototype configuration can be exchanged across user classes and functional disciplines. Success also depends on the degree to which existing, "off-the-shelf" M&S components can be rapidly reconfigured and linked into integrated virtual environments for evaluating the current product and process design and (if necessary) generating viable alternatives. Although it is possible to develop virtual prototyping environments in a more stovepiped manner, the risk and cost penalties associated with manual (or nonexistent) data transfer and translation between incompatible tools or simply "building from scratch" for each new application are considered to be prohibitive. This again highlights the importance of automated data transfer and software reuse/interoperability mechanisms within the architecture.

The following sections describe a proposed systems architecture for SBA applications. The architecture is not intended to specify any particular implementation strategy, but is rather intended as a generalized, high-level framework for enabling fundamental operational concepts and for establishing boundary conditions on potential implementations. This architecture specification is intended as a baseline, to be evolved and enhanced as more detailed, domain-specific issues of particular user communities are investigated and resolved.

Top-Level Architecture View

An approach that is frequently used to describe highly complex systems is to first describe the system at a high level of abstraction and then hierarchically decompose the system into greater levels of detail. This top-down approach was the strategy taken here, although only one level of decomposition is provided and, then, only for one of the three top-level architecture components. A depiction of the top-level view of the SBA systems architecture is provided in Figure 5-7. The three primary components of this architecture view are collaborative environments, a DoD/ Industry Resource Repository, and Distributed Product Descriptions (DPDs).

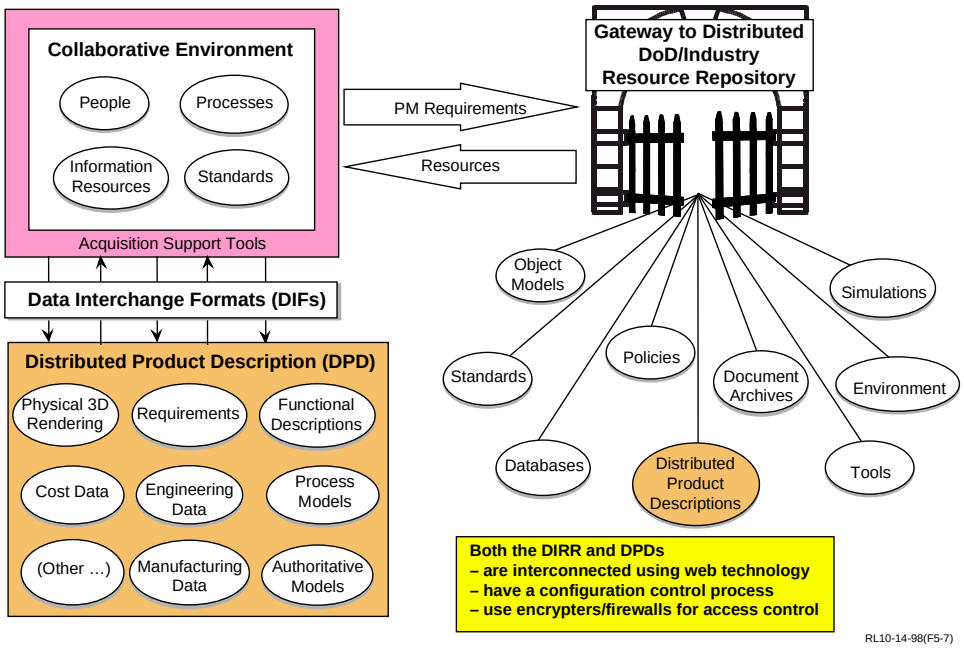


Figure 5-7. Top-level view of SBA systems architecture.

Collaborative Environments. The concept of a collaborative environment was first introduced in the SBA operational architecture description earlier in this report. Fundamentally, the intent is to align and associate teams of functional area experts and supporting sets of reusable, inter-operable tools and resources with particular classes of applications. For instance, product-focused collaborative environments would consist of personnel and hardware and software resources that typically address the design, development, and evaluation of products in that class (e.g., avionics, weapons, ground vehicles). Other types of collaborative environments may have an alternative focus, such as mission area. In any case, the primary purpose of a collaborative environment is to provide program managers with a reusable framework of tools, resources, and (potentially) IPTs within the bounds of a given application domain. This concept is more closely examined, from the systems perspective, in the *Collaborative Environment Reference Systems Architecture* sub-section.

DoD/Industry Resource Repository (DIRR). This component provides a web technology-based distributed repository of tools, information resources, and generic infrastructure components for use within and reuse across acquisition programs. Although individual collaborative environments are designed to contain and employ these types of elements within a specific context, a more comprehensive, general-purpose repository can facilitate reuse on a much broader scale. This repository can be considered to represent the union of the capabilities provided by all collaborative environments, providing a means for any individual collaborative environment to share local resources with other domains and for a collaborative environment to access and use external resources as required for new applications. The gateway to the repository provides a directory structure, browsers, and tailored commercial search engines to enable users to quickly discover resources (each described by a common metadata format) of user interest. Intelligent user interfaces provide extensive on-line assistance for employing the repository and

building complex search patterns. Firewalls and encryption devices are also included within the repository (as required) to support access and distribution of classified data and to enforce access restrictions on proprietary materials.

The data model underlying the repository is expected to be similar in concept to that of the DoD Modeling and Simulation Resource Repository (MSRR), but is employed on a much wider scale to support all enterprise functions. (The MSRR is a collection of modeling and simulation resources that are accessible through the World Wide Web (for unclassified) and Secret Internet Protocol Routing Network (SIPRNET) (for classified).) Standard metadata designations will be established to specify the sponsoring agency and accreditation status of each resource (as appropriate), thus facilitating determination of resource credibility and applicability. Other Government responsibilities include the development and enforcement of version and configuration control policies and procedures across the repository, and the identification and development of standards within classes of repository components (*see* Section 5.4, technical architecture description).

Distributed Product Descriptions (DPDs). A Distributed Product Description (DPD) is a distributed collection of product-centric information that is interconnected via web technology into what appears (to the user) to be a single, logically unified product representation. DPDs are composed primarily of three types of information: product data, product models, and process models. Product data specifies the characteristics of a product at any point in its development cycle, including requirements, program management data, cost data, engineering data, manufacturing data, and test data. Product models are authoritative representations of a product's behavior and/or performance. Process models are used to define the business operations necessary to define, develop, manufacture, deploy, and dispose of the product throughout its life cycle. DPDs may also contain other relevant product-related information, such as functional descriptions of product behavior and various categories of applicable metadata (e.g., VV&A status).

DPDs provide a common reference for Integrated Product and Process Development (IPPD) when coupled with appropriate automated support tools and a valid workflow model. That is, a DPD provides a common product view to all IPT participants at all times during the acquisition process, allowing simultaneous evaluation of the current product configuration from the perspective of each functional discipline. Thus, product designers can measure the performance of a product at the same time that product manufacturers evaluate the producibility of the product and logisticians assess the supportability of the product (assuming proper configuration management and version control). General-purpose viewers, web browsers, and search engines can be coupled to a DPD to provide IPTs with rapid access to all categories of information in the current product configuration and to quickly generate user-selectable views of the underlying data.

To the greatest practical degree, all information resident in a DPD is accessed according to standardized Data Interchange Formats (DIFs). The purpose of a DIF is to provide a common, intermediate format for describing product information. DIFs are derived from a common data model, which describes the semantics and syntax of shared data. These intermediate formats can be readily converted to any number of related output formats without loss or distortion of content using semi-automated, rapidly configurable parsing (adaptor) software. By defining an appropriate set of DIF standards, translators can be developed that allow any automated tools, implemented with a variety of technologies, to consume relevant product information without manual translation. Tools with various internal product data formats can also use translators to

generate and exchange product information with other tools or repositories via these same DIF standards, facilitating efficient, collaborative product development. Figure 5-8 illustrates this concept.

The implications of this strategy are significant from several perspectives. First, the selection of tool sets within a collaborative environment can be made independently of the native data formats used by the tools. This is because conformance to all applicable DIFs guarantees the ability to freely and directly exchange associated product information. Second, the costly and error-prone process of manual data translation is eliminated for all product data definable by DIFs, promoting tool interoperability. Third, since all collaborative environments operate from the same DIF standards, a tool that performs some function, via manipulation of DPD information, in one collaborative environment should be able to perform a similar function in a different collaborative environment without modification, thus facilitating reuse across application domains. This last concept applies within individual collaborative environments as well, as "plug and play" of different tool implementations become more achievable.

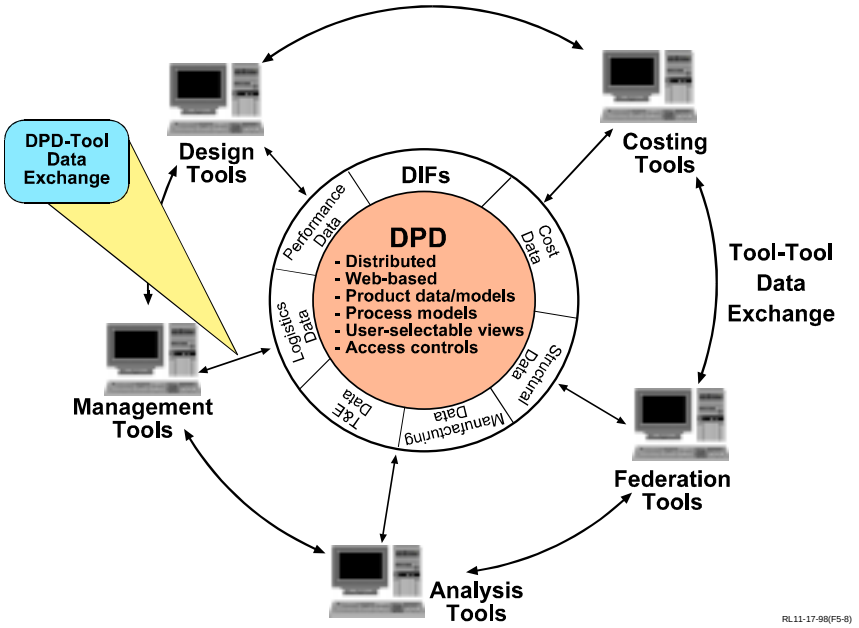


Figure 5-8. Distributed product descriptions (DPDs) and data interchange formats (DIFs).

Concept-of-Operations

The concept of operations associated with this high-level view of the SBA architecture is quite straightforward. As a new acquisition program is initiated, the program manager makes a determination as to whether the set of tools and resources provided by the appropriate collaborative environment is sufficient to achieve program objectives. If resource deficiencies are identified, the DIRR may be accessed and scrutinized for additional reuse opportunities. Once all required resources are identified and modified (if necessary) for use within the local collaborative environment, the tools and process models intrinsic to that environment are used to construct and evolve a virtual prototype through several design iterations. The product development process heavily leverages the DPD as a product view generator and integration mechanism across all

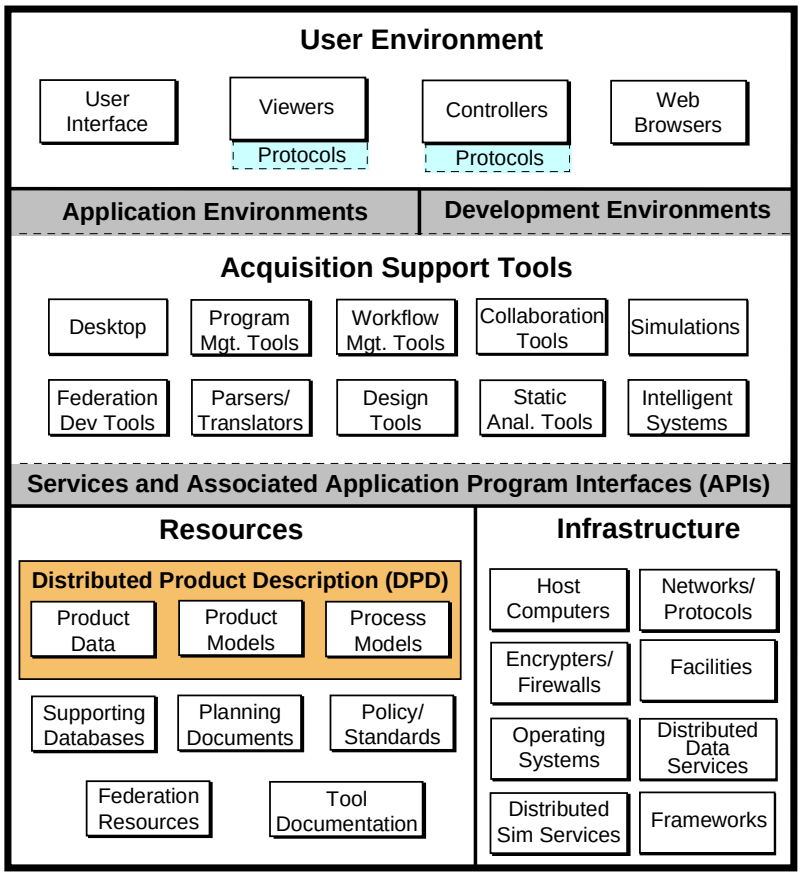
functional disciplines. As the product moves into production, the DPD and all other reusable products are archived locally and made available for broader reuse via the DIRR.

Although this top-level view of the architecture represents a significant abstraction from reality, it is useful for describing the basic technical approach. In the next section, a closer examination is provided of the inner workings of the collaborative environment component of the architecture.

Collaborative Environment Reference Systems Architecture

As previously stated, within the context of SBA, a CE is a long-lasting collection of interoperable tools and data bases, authoritative information resources, product/process models, and subject matter experts (SMEs) that are focused on a common domain or set of problems. All resources within a given collaborative environment are selected to maximize the overall effectiveness and efficiency by which applications are addressed within that domain. Although this type of flexibility is important, it also introduces the danger of stovepipes. That is, it is entirely possible for any given collaborative environment to focus and align its internal structure and supporting resources so strongly on a particular class of applications that cross-domain tool reuse and data sharing are impeded. For this reason, a reference systems architecture for all DoD collaborative environments is needed that globally defines the types of components that one can find in a collaborative environment, and (at a high level) how these components interrelate to one another. The primary intent of this reference systems architecture is to highlight the required "pieceparts" and opportunities for interface standards (in concert with the technical architecture) associated with a collaborative environment while providing developers with a high degree of flexibility regarding specific implementation approaches. For instance, some implementations of the architecture may be strongly object-oriented, using objects and object models to represent all product and process information, while other implementations may be more procedurally oriented. However the architecture is instantiated, the basic building blocks defined by this architecture should be present, as required, in all implemented systems.

The remainder of this section defines the basic system components associated with each layer of the architecture and highlights the technical challenges associated with developing and effectively using each component within and across collaborative environments. In no cases are any of these challenges considered unattainable, but all require investigation and research as highlighted in Section 7, Implementation Recommendations. An illustration of the Collaborative Environment Reference Systems Architecture is presented in Figure 5-9.



RL11-17-98(F5-9)

Figure 5-9. Collaborative Environment Reference Systems Architecture.

User Environment

This topmost layer of the architecture includes those system components that directly interface with the end users of the system.

Components

User Interface. The means by which the functionality inherent to a system is presented and made available to the user of the system. Well-designed, modern Graphical User Interfaces (GUIs) can "hide" much of the underlying complexity of a system from the user and can be used to tailor a system to user skill levels and specific use contexts.

Viewers. A means of providing a visual display of information resident in a system. In some cases, the viewer will be an integrated component of the support tools accessed via the user environment. In other cases, significant efficiencies can be gained by de-coupling the viewer from the underlying application. Under this type of software development strategy (Model-View-Controller, or MVC), the viewer and the application are separate components, with the flow of information from application to viewer governed by an established protocol. This allows (1) either the viewer or the associated application to be modified without requiring modification of

the other, (2) a single viewer to be reused across multiple applications, and (3) multiple viewers to be associated with a single application.

Controllers. A means of providing user command of applications in the system environment. Controllers are primarily a software function to handle user-initiated events, such as mouse interactions or keystrokes. Like viewers, controllers can be integrated directly into the support tools used by the system operator or they can be de-coupled to achieve certain efficiencies in the system implementation.

Web Browsers. A software application used to access information resources via the Internet. Many robust commercial web browsers exist (Netscape, Internet Explorer, etc.) that are inexpensive (or free) and easy to use.

Technical Challenges

At the level of the user environment, the most significant technical challenge is in the area of *visualization*. Certainly, virtual prototyping relies on the ability to visualize the appearance and behavior of a product within a synthetic environment. In addition, virtual manufacturing requires the ability to visualize factory layouts and shop floor processes and activities within a production facility, and training applications need a realistic depiction of the operating environment for which the product is intended. These requirements highlight the need for integrating affordable, high-resolution three-dimensional (3-D) displays with systems that produce a realistic virtual environment for studying the behavior and performance of a given product.

Although visualization technology is relatively mature, the immersion of humans in a virtual prototyping environment currently requires artificial tethers such as helmet-mounted displays and data gloves. These devices are often bulky and uncomfortable to use, making it impractical for a user to operate in a virtual environment for any significant length of time. Near term, technologies are needed that can reduce the weight and size of such devices. Longer-term, alternative approaches to human immersion in virtual environments are needed which will completely replace these devices, such as sensory aids that adjust the visual scene on the basis of changes to the orientation and physical location of the participant. Technologies that allow the immersion of several human participants at the same time and also provide improvements in the resolution, accuracy, and responsiveness of the visual scene also are needed to fully realize the benefits of virtual prototyping.[8]

Acquisition Support Tools

This next layer of the architecture includes those systems components that directly access, manipulate, and generate product information within a specific user context for the purpose of gaining knowledge and insight about the product.

In addition to the tool components described below, there are two optional “sub-layers” that can be defined at this level. First, it is recognized that, within a collaborative environment, there is more than one way in which the tool layer can be implemented. One strategy would be to implement each supporting tool in a stand-alone fashion, which minimizes dependency on external services and facilitates reuse within other collaborative environments. However, in situations where a broad suite of supporting tools can be developed and can operate from a core set of common services, it may make sense to implement the tool layer in a client-server fashion.

If this option is chosen, the common services would reside in a layer below the tools and would be accessed via an Application Program Interface (API) using an Object Request Broker (ORB) or other infrastructure component. Although the API need only be fixed within a given collaborative environment, cross-domain agreements on common APIs can significantly increase the potential of reuse of both services and client programs.

The other optional sub-layer at this level refers to the potential utility of tightly integrated application or development environments. These types of environments interconnect selected components from multiple layers into a highly cohesive framework focused upon a single use context. In an *application environment*, one might find a user interface, a web interface, several selected product design, analysis, and workflow management tools, an intranet of key resources (including DPDs), selected infrastructure components (data base management systems (DBMSs), postprocessors, etc.), and a manager shell all tightly integrated together under a backbone or "application engine" of common services. In a *development environment*, a similar set of tightly integrated components can also be pulled together under a common manager shell and backbone (through an API), with the environment in this case being specifically tailored to building new software components rather than developing a product. In both cases, these types of environments provide the user with a single system "look and feel," rather than simply applying the tools in a more autonomous fashion.

The number of examples of application environments in the DoD is small, but growing. For instance, the Joint Simulation System (JSIMS) can be thought of as an application environment. In the emerging JSIMS design, numerous application components (simulations; exercise planning, generation, and control tools; data bases; translators; common services libraries) are highly integrated and operate on top of a common infrastructure layer, for the specific purpose of military training and doctrine development. The Simulation Based Design/Distributed Knowledge Environment (SBD/DKE) concept of a Smart Enterprise Model (SEM) can also be thought of as an application environment. In a SEM, product data and product and process models are tightly integrated (under a common object model) with 3-D viewers (e.g., Virtual Reality Modeling Language (VRML)), simulations, design and analysis tools, and workflow and enterprise management tools, all within a "virtual operating system" called the SBD/DKE Core Processing System (CPS). With respect to development environments, many established commercial software development environments exist, such as the VisualAge Smalltalk environment currently in use within the JWARS program. The Joint Modeling and Simulation System (JMASS) is an example of an environment that encompasses both application and development elements since integrated services are provided for both simulation construction and configuration and for application control and post-processing.

Components

Desktop. An assembly of information systems components that provide basic office automation within a system environment. This generally includes such commercial components as word processors, spreadsheet programs, graphic tools, and desktop configuration utilities. These types of components are widely available at low cost from numerous office supply vendors.

Program Management Tools. A class of tools that provides automated support to program managers for monitoring and controlling a project. Capabilities generally include program planning, configuration management, resource allocation, and schedule, cost, and problem

tracking. These tools also implement the process models (in the Resources layer) that define standard business practices within the domain.

Workflow Management Tools. A special class of program management tools that are used to manage the sequence of activities and processes necessary to produce a product. This class of tools integrates and implements the process models resident in the Resources layer of the architecture to the degree necessary to fully operate an enterprise. These tools support the analysis of different aggregations of low-level design, development, and production processes in "what if" situations, and allow individual work packages to be dispatched and progress monitored across distributed IPTs.

Collaboration Tools. A class of tools that permits (distributed) teams to cooperate and work together toward common goals. Simple office automation, such as telephone, telefax, and e-mail provide a rudimentary means of collaboration. Video teleconferencing (VTC), groupware, and whiteboarding tools embody more modern collaboration technologies, with techniques such as natural language processing and distributed document control currently in research and development.

Simulations. An automated environment to study the behavior and performance of a system model in a temporal dimension. Simulations are used in the acquisition process for many purposes. For instance, simulations can be used to establish operational requirements for a product, to facilitate the development of early product design concepts and iterative evolution of the product design, to study and optimize the workflow processes that are used to manufacture the product, and to support operational training and long-term supportability. Simulations can be characterized according to their type (live, virtual, or constructive), the functional area they support, and the level of detail or fidelity the simulation provides. In an integrated collaborative environment, subsets of established, accredited collections of domain-specific simulations can be rapidly configured together in different ways (using HLA) to address issues relevant to different stages of the systems acquisition process.

Federation Development Tools. Any type of automated tool used to support the development and execution of HLA federations. Some tools in this class are unique to HLA federations, such as HLA object modeling tools and federation testing tools. Other federation development tools are general-purpose distributed simulation tools, such as scenario generators and data collectors. Still other federation development tools provide general support for distributed software development, such as configuration management tools and requirements traceability tools. Although some tools in this class are Government-sponsored, a commercial marketplace for federation tools is rapidly developing.

Parsers/Translators. A software utility that converts the data representations native to one system to the internal data representation of another system. When data that are commonly exchanged between different tools or between a tool and a data repository are defined according to standardized schemas and interfaces (data interchange formats), parsers provide the reusable "middleware" that is necessary to de-couple the specific tool implementation from modifications in the DIFs. Other types of translation middleware components may also be useful in certain types of applications, such as "wrappers" for legacy model/data base integration.

Intelligent Systems. The application of artificial intelligence (AI) and other nonlinear solution techniques to provide automated "expert" advice to the user of a system. The primary purpose of this type of component is to quickly post-process and analyze the potentially large amounts of data produced by a supporting tool (or suite of tools) into a form that provides immediate, useful information to the user. Knowledge-based systems can also help users determine the adequacy of the tools and resources within the system environment to address certain types of program requirements and to guide users to existing external resources that can be reused and reconfigured to support their immediate needs.

Design Tools. A class of tools that support iterative, spiral product design. In present-day acquisition programs, these types of tools are primarily used to support 3-D modeling of product design concepts (e.g., CATIA$_R$, AutoCAD) and factory layout options. In some cases, the output of these tools may be used to drive virtual prototyping and virtual manufacturing simulations. In future acquisition programs, these types of tools can be tightly integrated with models and simulations of other functional disciplines (cost, training, logistics, etc.) in order to address all aspects of product development early and simultaneously during product design.

Static Analysis Tools. A class of tools that allow the examination of product behavior and performance (like simulations), but without a temporal dimension. For instance, tools that implement operations research techniques to analytically address certain types of resource allocation problems would be in this class of tools. Cost and economic modeling could also be included here. Static analysis of 3-D product models (line-of-sight testing, interference testing, etc.) would also be conducted via this class of tools.

Technical Challenges

It is at this level of the architecture that the greatest technical challenges present themselves. Many of these challenges reflect a need for supporting standards and thus will be discussed in Section 5.4, To-Be Technical Architecture. It is also true that there are many well-known M&S technical issues that are not unique to acquisition programs. Several of these types of issues have been highlighted in the previous section on current M&S shortfalls. The issues discussed here are those that are not related to standards development and are critical to achieving the full set of functionality described by the tool components in this layer.

The primary technical challenges can be described according to three main categories:

Interoperability and Reuse. The root of this issue lies in the very wide range of different models and simulations in use within the DoD. Historically, models and simulations have largely been developed in a stovepiped fashion, with the requirements of individual applications driving such design characteristics as level of fidelity, data format, language, time management approach, and supporting hardware. The existence of these types of diverse characteristics makes achieving interoperability between M&S systems very difficult. Although the DoD High Level Architecture (HLA) and the other components of the Common Technical Framework represent a major step toward achieving the vision for interoperable, composable systems (as described in the DoD M&S Master Plan), it is clear that many technical issues still remain. For instance, achieving interoperability between models and simulations at different levels of aggregation has been and continues to be a particularly vexing problem. Technologies and methodologies to address this problem are particularly important for acquisition applications because product models of various degrees of maturity need to be able to interact consistently in virtual

environments. Another interoperability-related issue pertains to the way certain types of tools interact with simulations. For example, how do users of CAD/CAM systems employ simulation to quickly evaluate the performance implications of modifications to product material or structural qualities? Additional M&S interoperability issues, such as inconsistent semantics of common terms and incompatible representations of the natural environment, also impede the ability to reuse existing M&S components in large-scale distributed applications.

Costing Tools. Traditional cost models are highly parametric in nature and cannot provide the detail and accuracy required to support design trade-off decisions during product development. More advanced cost estimation tools are needed that can directly use the information in design and manufacturing data bases in order to quickly and accurately evaluate the cost impact of product and process data modifications. The development of such models will help to accelerate the trend toward more modern approaches to cost accounting.

AI Tools and Techniques. The need from the acquisition perspective is threefold. First, since product data bases can be very large and difficult to navigate, intelligent systems are needed that can quickly locate and distill large amounts of distributed product information into a more concise, understandable, context-sensitive data view. Second, intelligent systems are needed that can "understand" the impact of design changes on all aspects of an enterprise. For example, intelligent systems could be used to determine how a given set of proposed product modifications (for instance, to improve performance) might impact other areas (e.g., logistics and costing) and could directly affect the enterprise workflow model to investigate these potential impacts before changes are implemented. Finally, improved representations of human behavior are needed that can more accurately reflect the effects of stress, fatigue, and morale on human performance in modern battlefield scenarios. In all cases, advances in machine intelligence are needed to achieve these needed capabilities.

Resources

This next layer of the architecture provides the information resources that are necessary to conduct an enterprise. This primarily includes reusable libraries of supporting documentation resources and reusable data resources for product design and development.

Components

Product Data. Any type of information that describes the current state of a product specification at some point in the systems acquisition process. This would include requirements data, engineering data, cost data, manufacturing data, logistics data, and whatever other types of data are required to fully define the product. This information is captured and made globally and instantly accessible to all members of distributed IPTs via DPDs.

Product Models. Authoritative representations of product behavior and performance. Each product model identified in a DPD can reference an actual software implementation of the product (data and methods) that has been developed to operate in a specific static analysis tool or dynamic virtual environment. For instance, a single DPD for a radar system might reference several different product models, each of which is intended for use in different simulation systems (JWARS, JSIMS, JMASS, etc.). Alternatively, product behavior may also be represented via appropriate algorithms, which have not been implemented in software. Each product model is

based on a common functional and operational description (included in the DPD) that provides the basis for verification and validation of the model. The results of V&V testing and the level of sponsor accreditation currently associated with the model are additional categories of product data included in a DPD.

Process Models. A depiction of the processes and activities relevant to operating an enterprise. For instance, the specification of design processes are necessary to fully define the systems engineering approach to be used to iterate and mature the product design over multiple cycles. The specification of manufacturing processes are necessary to define the low-level procedures needed to fabricate and assemble a product and also to enable the identification of appropriate aggregations of these low-level sub-processes that together specify the overall flow of control on the factory floor. Process models for test and evaluation (e.g., STEP), operational support, VV&A, and standard business practices are also necessary to fully define an enterprise. A wide range of tools may apply these process models for the purpose of optimization and implementation.

Supporting Data Bases. Reusable data stores that are used in the acquisition process but that are not integrated into the DPD. Examples include scenario data bases, natural environment data bases, knowledge bases, data element dictionaries, and accredited functional descriptions of military operations.

Planning Documents. A library of potentially reusable documents that describe the plans that were developed to support the design and development of products within the associated domain. Examples of planning documents include system engineering (workflow) plans, VV&A plans, security plans, test plans, and configuration management plans. An appropriate set of these documents may provide a more efficient starting point for new product development than redeveloping these supporting documents for each new application.

Policy and Standards. A library of the policy directives and associated standards that apply to the development of products within the given domain. In all cases, the policies that drive business practices and the standards that drive product workflows should be tightly integrated into the process models exercised by program and workflow management tools. However, this separate library needs to be maintained as a reference for the development of new process models.

Federation Resources. A library of supporting resources that are used exclusively in the development of HLA federations. For instance, libraries of HLA Simulation Object Models (SOMs) for federates within the application domain might be archived for the purpose of quickly determining the appropriate configuration of simulations to address a particular problem. Existing Federation Object Models (FOMs) and highly reusable subsets of common (persistent) FOMs can be used to facilitate new FOM development and, perhaps, to drive the interfaces of new federates. Libraries of common nomenclature for federation data elements and documents that describe potential federation implementation approaches also represent reusable federation resources.

Tool Documentation. A library of user and technical reference manuals that document each component in the Acquisition Support Tools layer. This information also extends to the resources

and infrastructure used by the tools to accomplish their function. Appropriate documentation is needed to apply the support tools in a knowledgeable and informed manner.

Technical Challenges

As in the previous layer, there are many technical challenges that must be addressed to develop a full set of resources for acquisition programs. However, those technical challenges that can be best met through standards development will be discussed Section 5.4, To-Be Technical Architecture.

Several technical challenges relate to how the DPD concept can be achieved. As described earlier, DPDs are composed of product data, product models, and process models. From the product data perspective, the main technical issue is how *global consistency* is maintained across a rapidly evolving distributed product data base and how IPTs coordinate and control the implementation of proposed product data modifications. From both the product and process model perspective, the primary technical issue is a *general lack of reusable model components* from which higher-level system representations can be developed. That is, although the architectural structure supports the concept of composable systems, many of the "pieceparts" required to develop such systems do not currently exist. For instance, an immediate need has been expressed in several user communities for better physics-based environmental models that can accurately represent dynamic environmental changes and product–environment interactions in virtual worlds. Another example is in the manufacturing area since generic models of low-level shop floor processes are needed to more accurately represent factory operations and alternative manufacturing approaches. Examples of other necessary DPD components that are presently immature and/or incomplete include established, affordable processes for product and process model validation and recognized, authoritative representations of the battlespace that pertain to the product domain (i.e., need for continued CMMS development).

In addition to technical challenges directly related to DPD development, two other related technical challenges can be identified at this level of the architecture. The first of these challenges refers to how intermediate forms of product information are expressed and become captured in a DPD. This is often referred to as propositional knowledge. The basic issue is how proposed additions and/or enhancements to an evolving product description are introduced and negotiated within a collaboration of distributed IPT members, how such information is represented in its intermediate form, what tools are required to manage this information, and by what process does this information become a configuration-controlled element of a DPD. The second technical challenge relates to how data that populates supporting knowledge bases (*knowledge acquisition*) is captured and how such knowledge should be represented so that intelligent systems tools can reason on it. This is an important area of research and general debate in the AI community, but one that must be addressed to enable the intelligent tools discussed in the previous layer.

Infrastructure

This layer of the architecture provides the basic equipment (hardware and other facilities) and low-level software mechanisms that enable basic enterprise operations and the processing and exchange of product information within a collaborative environment.

Components

Host Computers. The computing platforms used in the acquisition process. These can range from simple desktop computers to the mainframe array processors needed in some types of physics-based modeling and simulation.

Networks and Protocols. The physical local area networks (LANs) and wide area networks (WANs) used to interconnect distributed computer systems, and the form by which information is transferred "over the wire." Examples of protocols include Transmission Control Protocol/Internet Protocol (TCP/IP) and User Datagram Protocol/Internet Protocol (UDP/IP).

Encrypters and Firewalls. Encrypters are devices that codify and de-codify classified data that are being transferred between different systems. The coding process maps a readable file into an unreadable file and requires a decryption key to translate the file back to a readable form. Thus, even if transmitted files are intercepted, they will be unintelligible. Firewalls are mechanisms for shielding host machines or sub-nets from unwanted activity initiated externally. Although firewalls are primarily intended as a network security mechanism, they can also restrict unauthorized external access to proprietary product information.

Facilities. The physical assets needed to host a complete systems acquisition environment. Facilities include buildings (or floor space within a building), the building infrastructure (power, lighting, air conditioning, etc.), and whatever equipment is need to support the enterprise (such as fabrication tools and test equipment).

Operating Systems. The means by which the physical resources of a host computer are made available to a user. The operating system provides an abstraction layer so that computer users do not have to know or understand the low-level operations that need to take place to store files, retrieve files, build directory structures, etc. Distributed operating systems pool the physical resources of multiple networked computers to present to the user what appears to be a single, very large, and powerful computer system. Many commercial distributed operating systems employ the Common Object Request Broker Architecture (CORBA) as a standardized framework for implementing distributed object services.

Distributed Data Services. Software services and mechanisms to manage access and storage of distributed data within an enterprise. Collectively, these services provide a distributed DBMS, accepting complex user queries and integrating multiple data sources, with potentially diverse data formats, into a common, user-selected data view.

Distributed Simulation Services. Software services that allow multiple interacting simulation systems to interoperate in a meaningful way. Examples of distributed simulation service groups include object management, time management, data distribution management, and federation management. A full set of distributed simulation services are specified in the HLA Interface Specification and implemented in the HLA Runtime Infrastructure (RTI).

Frameworks. Sets of object classes that interrelate, interact, and collaborate to carry out a given set of responsibilities. Frameworks provide a foundation for new application development by providing a base set of functionality for a particular problem domain that can be customized through sub-classing (instead of modifying the original code) to achieve the full desired capability. Frameworks facilitate rapid software development by allowing program designers to

reuse existing, validated software components to partially implement their system and thus focus their attention and energies on the particulars of their individual application. Most existing frameworks are highly generic in nature, providing items that are common to all programs with GUIs (e.g., Microsoft Foundation Classes). However, frameworks can also be built that are domain specific, providing a base set of functionality for both business and technical applications.

Technical Challenges

At the infrastructure level, many of the stated components are already quite mature, although the never-ending quest for more network bandwidth and more powerful computers continues to challenge the commercial sector. Training applications are continuing to drive many scalability requirements, as simulation exercises consisting of tens (or even hundreds) of thousands of battlefield entities need to be supported in distributed, heterogeneous computing environments. Requirements for *multi-level security* (MLS) are common in these types of environments, presenting technical challenges for new distributed security architectures and advanced encryption techniques. Requirements for *interfaces with live systems* are also common in such exercises, indicating a need for new tools and techniques to facilitate seamless integration of live and virtual systems. Further advances in *computing power* are required in the virtual prototyping area, as high-resolution visualization environments, linked to highly detailed physics-based models and CAD applications, are needed for real-time analysis of product behavior and performance.

Widespread use of DPDs and (on a broader scale) the DIRR also implies at least two significant technical challenges. First, in addition to support for MLS, reliable, *robust firewall systems* must be developed and demonstrated to install confidence that proprietary information can be protected in an open environment. Although this is more of a cultural barrier than a technical one, such confidence is critical for an open DoD/Industry repository to operate effectively. Second, advanced *distributed data services*, which can manage the assembly of distilled, context-sensitive data views from multiple distributed data bases based on user-initiated complex data queries, are needed to fully implement the DPD concept.

5.4 TO-BE TECHNICAL ARCHITECTURE

This section provides a notional view of the standards and local conventions that govern the implementation of collaborative environments. As discussed above, these collaborative environments must conform to the “to-be” view of the SBA systems architecture.

Purpose

Section 5.3 has described a future vision for the types of systems components needed to instantiate a collaborative environment, included the high-level interrelationships that exist between these component classes. As discussed earlier, the fundamental goal of the SBA systems architecture is to facilitate data transfer and software interoperability and reuse both within and across acquisition programs. Such goals can be at least partially addressed via mechanisms that exist within the systems architecture, such as network services, frameworks, and other similar infrastructure elements. However, it is important to recognize that these mechanisms simply enable information sharing and reuse, and that appropriate sets of supporting standards must be defined to augment and complement the SBA systems architecture for these stated objectives to

be realized in practical applications. For example, while distributed data services are essential to integrating multiple, physically dispersed data sources into a unified data environment, the specific data view presented to the user may need to be redefined for each new application if appropriate formatting standards do not exist. Standards are also necessary to provide practical bounds on the specific tools and resources that will be made available within a given collaborative environment, and for the processes and procedures that define how these components are applied to practical applications. For tools, resources, or infrastructure components that may be common to all collaborative environments, establishment of broader DoD standards can also strongly facilitate reuse, interoperability, and information sharing.

The primary purpose of this section is to identify the rules, standards, and conventions that must be associated with all implementations of collaborative environments. Although there are no real "showstoppers" or barriers to implementing the technical architecture, there are several technical challenges associated with developing and implementing an appropriate set of DoD and community standards. These challenges will be identified and discussed in this section.

Technical Architecture Description

DoD Standards

As was mentioned in the earlier "as-is" view of the SBA technical architecture, the evolving DoD Joint Technical Architecture (JTA)[40] currently mandates the minimum set of standards and guidelines that are necessary to facilitate interoperability among systems that use information technology. As such, as the DoD JTA continues to mature, it is expected that many of the standards specified in the JTA document will apply to the development of new components (and fully integrated systems) within collaborative environments. For instance, the information processing section of the JTA specifies explicit standards for data management services (Structured Query Language), document interchange (Hypertext Markup Language), and graphics interchange (Graphics Interchange Format). Other example mandates include the use of TCP/IP and UDP/IP for transport services, and the Defense Data Dictionary System (DDDS) for standard DoD terminology. In addition to mandates that apply to all DoD information systems, the JTA also specifies standards that apply only to specific domains. The standards defined in the Modeling and Simulation Annex to the JTA are expected to be especially important to SBA applications. Currently, the M&S specific standards are primarily focused on the DoD HLA.

There are also numerous references within the JTA to the concept of a standardized Common Operating Environment (COE), which provides a set of modularized software services that are consistent with the DoD Technical Reference Model (TRM) and which are accessed by higher-order applications via a stable set of APIs. The Defense Information Infrastructure (DII) COE represents an implementation of the COE concept, providing a "plug-and-play" open architecture designed around a client-server model.[27b] Although use of the DII COE is currently limited to C4I applications, it is quite possible that DII COE based application environments will be prevalent within collaborative environments in future years. Implementations of the SBA systems architecture can fully support the COE concept through proper access (via common APIs) and use of COE segments resident in the "services" layer. Although this layer is presently optional, broad community migration to this fundamental approach to information systems architecture may necessitate instantiation of this layer in most collaborative environments.

Since the JTA is fully intended to apply to the types of information systems that typically support acquisition, the JTA can be considered to be the SBA technical architecture from the cross-collaborative environment perspective. Of course, not all of the mandated and/or emerging standards specified by the JTA will apply to all collaborative environments. However, whenever a given functionality is required within a collaborative environment, the core body and relevant domain annexes of the JTA should be examined for applicable standards.

Because the JTA identifies numerous constraints on how collaborative environments are developed and used, it is critical that suitable representatives of the acquisition community (particularly in the M&S area) become active participants in the continued development of the JTA. For instance, broad standards for product data templates and associated DIFs are strongly needed by the acquisition community in order for supporting tools to be reusable and interoperable across acquisition programs and shared by different collaborative environments. Such standards are minimal in the current version of the JTA, but existing efforts such as STEP (Standard for the Exchange of Product Model Data) can be leveraged to build the full set of required standards. There may be other opportunities for new JTA standards as well in the areas of federation resources (e.g., Object Model Data Dictionary (OMDD)), supporting data bases (e.g., threat, environmental), process models (Simulation, Test, and Evaluation Process), and possibly others. Without such active involvement in the development of DoD standards for acquisition applications, reuse of tools and resources both within and across collaborative environment boundaries will be significantly impeded.

Community Standards

In addition to standards that are DoD-wide, there are many standards that can be defined that are local to each collaborative environment. These types of standards or local conventions should be defined whenever the constraints they introduce facilitate rapid and efficient application development and utilization within the collaborative environment (and no broader DoD standard exists). For instance, the specification of the assets that should populate each required component of the collaborative environment can be thought of as a local standard. In other words, explicitly identifying the simulations that will be used within the bounds of the collaborative environment, the design tools, the data bases, the facilities, etc., are all decisions that need to be made at the level of the individual domain. Rules that govern how the hardware/software assets in the collaborative environment are applied to domain-specific applications are also defined local to the individual user community. Although these types of standards are as much a part of the overall SBA technical architecture as the broader DoD standards, it is not possible to anticipate in advance how each user community may define its local conventions. Thus, for the purpose of defining the SBA technical architecture, it is sufficient to say that the architecture encompasses the JTA standards that are relevant to the domain of the collaborative environment, and all local standards and conventions necessary to fully support the acquisition process in an efficient and robust manner. Organizations such as the Simulation Interoperability Standards Organization (SISO) provide useful forums for Industry/Government collaboration on such standards.

Technical Challenges

The most significant types of challenges associated with the SBA technical architecture are not technical in nature. Rather, the main challenges associated with developing and enacting an

appropriate set of cross-community standards and intra-community rules and conventions are related to getting disparate groups of users, who are used to working in a stovepiped fashion, to agree to migrate to a set of tool, resource, and/or infrastructure standards that are different from the way they commonly do business. This is a significant issue and one that requires commitment on the part of the potential users of the standard to participate in its development and to embrace and adopt the resulting product. A related issue is how to organize and structure the various standards development processes so as to achieve adequate buy-in from all potential users. Although these challenges are more cultural or process-oriented than technical, they are critical to instantiating the technical architecture.

The most significant technical challenge related to standards development pertains to DIFs. Specifically, the SBA systems architecture relies heavily on the existence of standard templates and DIFs to describe the various categories of product data. The existence of standard DIFs is critical for automated exchange of product information between different acquisition support tools and to facilitate interoperability of tools and reuse of information resources across acquisition programs. However, whether all types of product data are definable in standard DIFs is an open question. For instance, for any given product category, can DIFs be defined that capture all possible degrees of aggregation that may be required during product development? Another issue is how many unique DIFs will be required. That is, can a single DIF for sensor characteristics (as an example) be defined, or does each unique type of sensor require a separate DIF? Finally, are there types of product data that cannot be defined via standard DIFs, and if so, how are the tools that process these data classes reused in subsequent applications? Clearly, experimentation will be necessary to determine the extent to which the DIF concept can be applied.

5.5 OPPORTUNITIES FOR/FACILITATION OF REUSE

A 1996 DoD advisory committee defined reuse as the "use of knowledge or artifacts from existing systems to build new ones."[28] The need for reuse of data, documents, models, simulations, facilities, network infrastructures, and processes within and across acquisition programs is a fundamental tenet of SBA. Indeed, the extent to which information and artifacts are shared within and between acquisition programs can be considered a critical measure of success of the to-be SBA architecture. The benefits of reuse are life-cycle cost savings, shortened acquisition schedules, lower risk, and enhanced flexibility, interoperability, and credibility of simulation resources, data bases, and design tools.

The previous sections described the to-be operational, systems, and technical SBA architectures (Sections 5.2, 5.3, and 5.4, respectively) necessary to facilitate reuse in future DoD and Industry product designs. For clarity, this section merges and summarizes the opportunities for, and facilitation of, reuse in the future SBA acquisition environment. The following subsections briefly summarize who will do the sharing, what information and artifacts they will share, and what enablers are necessary to promote sharing.

Collaborative Partnerships

The establishment of collaborative partnerships is essential because these define the people, organizations, and communities that will share information and artifacts in the SBA environment.

Section 5.2 (To-Be Operational Architecture) describes multi-tiered collaborative environments that will promote the sharing of M&S resources, expertise, and data among acquisition stakeholders across functional and program boundaries. Customers tapping into the collaborative environment at each level will have ready access to product data, design tools, computers and facilities, simulation resources, test data, and documentation.

The collaborative environments will be composed of whatever participants make the most sense. It would be a mistake to define fixed structures and organizational relationships that cannot grow, evolve, and spawn new reuse partnerships over time. The Task Force expects that once a general framework for reuse is established (i.e., collaborative environments) and a minimum set of standards is defined to facilitate reuse within and across acquisition programs, multi-tiered and cooperative reuse partnerships will flourish throughout DoD, Industry, and Academia. New reuse partnerships will be established over time, and existing ones will evolve and begin sharing between them.

The most benefit is expected from collaborative environments among mission and product areas because these partnerships will promote reuse across acquisition phases, over entire product life cycles, and between product designs. Collaboration among product areas will ensure that experts from the various functional disciplines of acquisition, such as systems engineering, manufacturing, test and evaluation, training, etc., share and reuse knowledge and artifacts related to the product design, production, test, and operational use of the product over the entire product life cycle. Collaboration within mission areas will ensure that knowledge and artifacts developed for one acquisition program are transferred to later programs and ensure that new technologies, concepts, and systems are developed in an operational context.

Knowledge and Artifacts

Participants in collaborative environments will share and reuse knowledge and artifacts. The scope of reuse in these environments will be much broader than simply the sharing of simulation tools and code among participants. Reuse in collaborative environments will cover the entire spectrum of sharing possibilities, from the reuse of algorithms, code, and simulations on one end of the spectrum, to the sharing of data, facilities, documents, people, and expertise on the other end.

There are many levels of reuse. Reuse can be as straightforward as the sharing of threat data and models between acquisition programs and functional disciplines. It can be as simple as the sharing of “lessons learned” between functional experts participating in collaborative environments across product designs but within a common mission area. Reuse can also be as complex as common developmental frameworks (e.g., the J-Series of models) and the reuse of model components via distributed simulation (e.g., HLA confederations). The to-be SBA architecture will facilitate all levels of reuse. It will bring the people, processes, information, tools, and facilities together in a common environment.

Section 5.3 (To-Be Systems Architecture) defines three types of knowledge and artifacts that will be shared within collaborative environments: resources, infrastructure, and acquisition support tools.

- Resources are the information and tools necessary to conduct an enterprise, including product data, product models, process models, and the supporting data bases, documentation, policy/standards, and federation resources.

- Infrastructure is the basic equipment (hardware and other facilities) and low-level software mechanisms that enable basic enterprise operations and the processing and exchange of product information between collaborative environment participants. Infrastructure components include host computers, encrypters/firewalls, operating systems, distributed simulation services, networks/protocols, facilities, and distributed data services.
- Acquisition support tools are components that directly access, manipulate, and generate product information. They include desktops, program management tools, collaboration tools, simulation frameworks, federation development tools, parsers/translators, design tools, static analysis tools (like cost models), and intelligent systems. The J-Series models are simulation frameworks that will provide common operating environments for product models having a spectrum of detail/fidelity and runtime characteristics. For example, the training community needs product representations that run in real time and focus on providing realistic responses to human inputs, whereas the systems engineering community needs high-fidelity product models running faster than real time and with less focus on human interactions.

The DoD/Industry Resource Repository will contain global knowledge and artifacts appropriate for all sharing and reuse by all local collaborative environments. It will provide a means for any individual collaborative environment to share resources with other domains, and for a collaborative environment to access and use external resources as required for new applications. It will contain, among other things, a minimum set of standards and protocols necessary to facilitate the sharing of information and artifacts between collaborative environments, standard threat models (data and behavior), scenarios, authoritative representations of natural environments (e.g., oceans, atmospheres, terrain, etc.), and other general-purpose data bases in standardized formats. As products move into production, the product data, product models, process models, and other reusable information and artifacts will be made available for broader reuse via this global repository.

Within the to-be SBA architecture, frameworks can be employed to facilitate the reuse of software code and algorithms among participants of collaborative environments. Frameworks are sets of foundation software modules that interrelate and interact to model entity characteristics and behaviors. They are object classes in object-oriented software designs. Frameworks will facilitate rapid software development by allowing software engineers to reuse existing, validated software components in their designs. For example, a collaborative environment might have a framework for a generic class of missile objects. Software engineers needing to model a new interceptor might use the missile object framework as the foundation for the new interceptor model, inheriting the missile movement, mass property, guidance, and propulsive features already built-in to the foundation object.

Enablers

Sections 5.3 and 5.4 (To-Be Systems and Technical Architectures) fully describe the enablers necessary to facilitate the open, free sharing of knowledge and artifacts within and between collaborative environments. Primary among these are the following:

- Multi-level collaborative environments to align and associate teams of functional area experts and that support sets of reusable tools and resources with particular classes of appli-

cations and provide a “one-stop shopping” resource for program managers to conduct and direct their programs.

- A DoD/Industry Resource Repository to facilitate the sharing and reuse of M&S resources and other design tools, data, and standards needed by all local collaborative environments.
- Use of Distributed Product Descriptions (DPDs) by all collaborative participants. DPDs are distributed, web-based repositories of product-centric information, including product data, product models, and process models.
- Data Interchange Formats (DIFs) to store and access DPD items. The purpose of a DIF is to provide a common, intermediate format for describing the semantics and syntax of shared data.
- Parsers and translators to convert, via manipulation of DPD information, data representations native to one system to the internal data representation of another system.
- Distributed services, networks, and standards (e.g., HLA) to facilitate distributed sharing and reuse between members of geographically distributed IPTs.

5.6 MODULE OWNERSHIP

The purpose of this section is to define ownership of each module of the Collaborative Environment Reference Systems Architecture and the responsibilities of module ownership.

Definition of Ownership

Module owners of the Collaborative Environment Reference Systems Architecture are defined as the DoD components who have formal responsibility for the modules. In most cases, this responsibility is derived from existing responsibilities for representations of military systems, data, and other functions. For example, the Military Services are responsible for digital systems representations of their Service-unique systems, while DIA and other components of the intelligence community are responsible for representations of non-U.S. systems and capabilities. Formal designation of the owner of each module of the Collaborative Environment Reference Systems Architecture facilitates implementation of individual collaborative environments and resources used in those collaborative environments. The owner is responsible for ensuring the development and validation and verification (V&V) of product models, data, simulations, and other design tools. Although the owner is responsible for development and V&V, the user is responsible for accreditation of the comprehensive product or process model(s) that is used in each collaborative environment.

Determination of Ownership

Ownership responsibility needs to be assigned to ensure that all members of collaborative environments and other stakeholders can collaborate effectively. Generally, ownership of modules is designated where commercial practices are not sufficiently robust to support products and applications for common and general use. Where appropriate commercial products exist and can be used to implement collaborative environments and architectures for those environments, it is not necessary to assign ownership.

There are several reasons to formally designate module owners:

- Formal ownership clarifies authoritative representation of DPDs in a collaborative environment. Currently, product models of the same systems are built by different organizations, and leaders of collaborative environments must know which organizations have formal responsibility for representations.
- Configuration management of models is facilitated by designated ownership. Numerous models have different variants maintained by different developers. A module owner would be the responsible organization for maintaining configuration of DPDs for specific purposes.
- VV&A is facilitated by formal ownership, with the owner of a model responsible for V&V of a model or DPD and multiple owners responsible for V&V of integrated simulations. Owners are also responsible for assisting users with accreditation.
- Owners are also responsible for developing DIFs and other standards. In most cases, this is a joint responsibility among different owners for DPDs for which responsibility is shared among different organizations and associated owners.
- Finally, owners are responsible for PPBS activities. Currently, selected model development activities with common and general use application, such as threat models, are funded by the initial organization that requires the product. Designated ownership would facilitate the funding of common and general use products by either a central funding source or a group of customer organizations.

Issues with Ownership Designation

The drawback to specifically designating a module owner is that module owners may not be funded to develop components of the architecture and associated product models. Often, the products are developed in response to customer requirements. A module owner cannot be expected to fund the development of common and general use requirements if that particular owner has responsibilities that are more immediate. For example, a frequent problem is that the first customer who requires a particular resource is expected to fund it, even though it has further applications.

Responsibilities of Collaborative Environments

Developers and owners of collaborative environments own particular instantiations of the Collaborative Environment Reference Systems Architecture for their individual collaborative environments. Accordingly, they are responsible for effective integration of the modules in the architecture and for ensuring that modules are represented and implemented correctly as part of individual instantiations.

When the Services have the same collaborative environment, the Services will need to designate a “champion” that would lead a multi-Service effort.

Designated Module Owners of the Collaborative Environment Reference Systems Architecture

The proposed owners of the modules in the Collaborative Environment Reference Systems Architecture are as follows:

Simulations, Product Models, Product Data, Support Data Bases

OSD agencies, the Services, and intelligence community organizations with DIA as the overall lead are, along with Industry, owners of models, simulations, and DPDs in accordance with their normal responsibilities. Critical to this concept is that the Collaborative Environment Reference Systems Architecture not deviate from existing organizational responsibilities. Hence, Services are responsible for representations of their own systems, while the DIA and other components of the Intelligence Community are responsible for representations of foreign systems and capabilities. This responsibility includes ensuring authoritative representations of DPDs and product models and how they are implemented in a simulation or a collaborative environment, and the verification and validation of those models, as well as assistance and support to the organization accrediting the representations.

Policy, Standards

The policy and standards modules have the same owners as the above group of modules pertaining to DPDs and the resources used to populate DPDs. USD(A&T) is designated the overall owner of SBA policy within the Department, with other organizations responsible in accordance with decentralized execution. The objective of this designation is to ensure a single DoD SBA policy, while giving the module owners the ability to implement the DoD policy effectively and efficiently.

Networks/Protocols, Facilities, Encrypters, Firewalls, Design Tools, Static Analysis Tools

The infrastructure on which the Collaborative Environment Reference Systems Architecture depends is owned by OSD and the Services, with the Defense Information Systems Agency (DISA), the National Security Agency (NSA), and Industry owning specialized portions. Implementations of the SBA Collaborative Environment Reference Systems Architecture should use existing infrastructures such as the Internet, Secret Internet Protocol Router Network (SIPRNET), and other components of the DII COE.