Modeling and Simulation Links with Command & Control (C^2) Systems

Considerations in Architecture Designs

Kevin Brandt
The MITRE Corporation
Joint Warfighting Center
PO Box 51037
Hampton, VA 23651-0037
757-726-6896
brandtk@mitre.org

Keywords:
Command and Control Systems, Modeling and Simulation Systems, C^4ISR,
Architectures, Design Criteria,
Mission, Security, Feasibility,
Coherence, Usability, Robustness, Affordability, Sustainability,
Modularity, Cohesiveness, Connectivity, Economy, Extensibility

ABSTRACT: *This concept paper explores objectives and requirements and proposes evaluation criteria for architectures linking command and control (C^2) systems with modeling and simulation (M&S) systems. It focuses on design requirements for data flows between Joint training models and mission applications or segments within the Defense Information Infrastructure Common Operating Environment (DII COE) including the DII COE Shared Data Environment (SHADE).*

1. Introduction

The paper opens with the rationale for establishing links between command and control (C^2) systems and modeling and simulation (M&S) systems. It then provides a brief review of the Joint-training mission domain. Next, it postulates design considerations for establishing data exchanges between M&S and C^2 systems. Afterward, it uses these proposed attributes to review an architecture prototype and to assess their impact on other designs and on end users in training exercises. As a by-product, it evaluates potential strengths and weaknesses of one prototype architecture. In conclusion, the paper explores potential implications for model designs and modular development within the DII COE.

However, this paper does **not** attempt to review methods for representing or modeling C^4ISR systems within simulations. Moreover, it does **not** develop operational architectures for specific use cases or training scenarios.

2. Rationale

Links between C^2 systems and M&S will sustain current uses and foster new applications. Today, connections between these domains drive computer aided training exercises. In the near future, automated links will provide embedded training environments for command, control, communications, computers, intelligence, surveillance, and reconnaissance (C^4ISR) systems. On the acquisition front, connections may stimulate testing of emerging systems over a range of operational missions and can provide extensive load tests (feasible with automated links between domains). In addition, tighter links can aid in course of action (COA) development and analysis. Automated links also foster mission planning and wargaming analysis. Moreover, they extend the planning environment to aid in the coordination and synchronization of mission tasks and actions for multiple subordinates. Over the long term, integrated links will enable full-scale mission rehearsals in mixed environments with constructive, virtual, and live forces.

3. Training and Mission Analysis

In the era of legacy training models and emerging semi-automated command and control systems, links between M&S and C4ISR domains are layered. In a traditional operational architecture, the training audience and mission analysts are in the top layer with limited direct access to the M&S tools – in the bottom layer – that are adjudicating synthetic events.

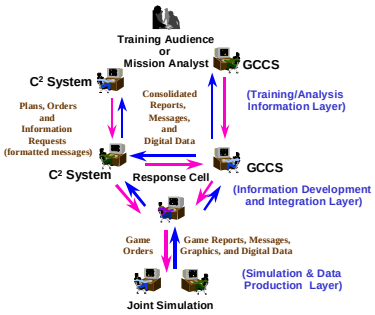

Figure 1: Layered Operational Architecture

Training audiences interact with supporting simulations through an information development and integration layer comprised of mission-area response cells. These cells translate training audience commands or requests for information and relay them to the appropriate supporting simulation. In addition, these cells consolidate and filter simulation-generated information and insert it into the target C4ISR system. Today some of these links are automated but most rely on manual intervention.

Future links between these domains may follow the traditional layered approach while seeking to automate services in the information development and integration layer. Alternatives seek to eliminate the need for this middle layer with fundamental changes in the supporting simulations and/or targeted C4ISR systems. Regardless of the approach, the quest for a fuzzy "seamless integration" fails to provide definitive criteria for architecture development. Within the C4ISR domain, the quest for the "seamless" grail has been supplanted by a more realistic move toward specified levels of interoperability.

4. Levels of Interoperability

Policies, technology, and resources can drive or restrain information architectures linking M&S with C2 systems. For example, links may be constrained by security policies that limit interconnections or they may be driven by a political dictum to connect to a coalition partner during a combined exercise. Similarly, resource constraints may drive long-term automation of tasks to gain efficiencies and improve response times or may limit network capacities and links available for specific events or within specific geographic areas. Likewise, technical approaches may present new obstacles while solving old problems.

In operational domains, requirements can limit the introduction of new technologies unless both evolve in parallel. For example, users now expect full-duplex data exchanges over tactical networks to core data servers. If architectures are to fully exploit global broadcast system and other wireless technologies, the technical requirement for full-duplex links must be eased. New technical architectures will be needed to enable parallel simplex pathways over multiple channels to fully exploit potentials for improved data distribution over wide areas. In addition, these changes will impact the development and application of new tactics, techniques, and procedures. Likewise, current links from M&S systems that exploit text-only message protocols will not suffice for long. As users' C2 systems expand to incorporate web-based video data streams, interactive, distributed collaborative planning systems, and robust data warehousing and mining technology, they will demand full-dimensional multi-media links.

As these simple examples show, the desired or required level of connectivity may be pushed or constrained by policy, technology, or costs. In contrast, actual connectivity is constrained by the realm of the practicable or the feasible.

C4I connectivity may be classified into one of the five categories developed by CISA to codify C4I systems interoperability. These five levels [1]: range from isolated (0) to connected (1) to distributed (2) and then to integrated (3) before culminating at universal (4).

While not explicitly developed to codify cross-domain links to M&S, the LISI metric provides a reasonable framework to scope the needed level of connectivity. In general, lower levels of interoperability require increased manual intervention to maintain links. However, higher levels of interoperability are not free. In general, they impose requirements for recurrent

[1] The C4I Integration Support Activity (CISA) has led the development of an extensive schema to codify levels of interoperability between C4I systems. Their Level of Information System Interoperability (LISI) metric provides the basis for an assessment of interoperability potential. While this scheme was not explicitly developed to address cross-domain links to models and simulations, the results are generally applicable to these cross-domain links.

coordination between independent programs, increased levels of engineering development, and robust configuration management.

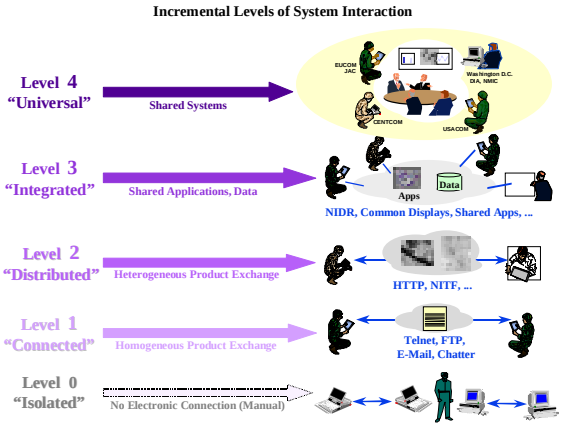

Figure 2: LISI Levels of Interoperability

Given the LISI model was not developed to classify links between the M&S and C[4]ISR domains, these connections may drive extensions to the LISI metric. Potential expansion of LISI warrants further study – especially in the classification of links between M&S components and standard gateways. Extension should allow the LISI metric to fully capture potential connections between systems built to common standards.

5. Standards and Specifications

Adoption of common standards promotes reuse of system architectures and component software. Emerging industry standards combine with evolving DoD specifications to dominate potential approaches. However, the rapid introduction of new technologies within the commercial sector must stimulate reevaluation of derived architectures to ensure past solutions continue to be efficient and effective.

Four pillars are poised to sustain the design of interfaces between C[4]ISR systems and M&S domains.

- The DoD Joint Technical Architecture (JTA)
- Defense Information Infrastructure (DII) Common Operating Environment (COE)
- DoD High Level Architecture (HLA) for Modeling and Simulation
- POSIX and Industry Software Standards

First, JTA compliance is a major consideration. If potential solutions do not comply, then interoperability with C[4]ISR components may suffer. If solutions fail to achieve interoperability with the targeted C[4]ISR systems, they will fail to achieve their primary objective. However, some interface challenges may be met only if the JTA is extended to incorporate new technologies. In these areas, interface developers should leverage the C[4]ISR standards development process to foster reuse of new engineering solutions.[2]

Second, given the DoD strategy to migrate most C[2]I systems into the DII COE, potential links with M&S stand to provide the greatest benefit by targeting these mission applications or other common DII COE segments. In addition, reuse of DII COE hardware, data standards, network infrastructures, and technical support personnel for M&S applications may be practical if equivalent environments are maintained for each domain. System reuse lowers resource requirements and promotes reduced maintenance, fielding, and training costs. Moreover, reuse can shorten development timelines. However, M&S applications and interface links must target DII COE Level[3] 5 compliance (or higher) to gain these benefits.

Third, in the M&S domain, DoD has promoted reuse by fostering development of standards-based links between simulations. Christened the high-level architecture (HLA), this standard defines required services in the runtime interface (RTI) between models within an HLA federation, sets metadata standards, and provides supporting tool sets. The RTI layer and services also provide a conduit to command and control systems. DoD policy stops short of mandating use of the HLA RTI as the single gateway to C[4]ISR systems. However, such an approach fosters reuse of links established for one component by other DII COE components or other mission applications.

Potential conflicts between DII COE guidelines[4] and HLA rule sets[5] should be considered in the development of technical architectures and software

---

[2] GCCS CM Policy, CJCSI 6722.01, 1 July 97.

[3] DII COE compliance is measured at 8 levels of conformance: levels 1 - 8. Joint agencies have established Level 5 compliance as the minimum acceptable level for Joint interoperability. Compliance has four measurement categories: runtime environment, style guide, architectural compatibility, and software quality. Compliance ratings are based on three factors.

1. The degree a system (or software segment) achieves in conformance to rules, standards, and specifications.
2. The degree of suitability that a segment has for integration into the DII COE.
3. The extent the segment makes use of COE services or duplicates them.

[4] DII COE guidelines push for reuse of services to avoid duplication of effort and to secure efficiencies.

[5] HLA rules specify a loose data coupling scheme for federates that does not appear to align with some of the more tightly-coupled DII COE SHADE concepts.

segments. However, these standards may need to be redefined and/or refined when M&S components are integrated into the DII COE.

Fourth, development to POSIX and other open standards enables portability of software modules across platforms (at source code levels). This flexibility supports placement of modules within the C4ISR domain even when the native operating systems cannot host all of the core components of the simulation. Moreover, these standards promote reuse of common libraries and services and provide well-defined interfaces. Thus, they can speed development, reduce maintenance, mitigate technical support burdens, and ease user-training requirements.

In addition to these four standards, several DoD directives or planning handbooks guide the development of integrated architectures and promulgate integration and testing requirements.[6]

6. Design Considerations

Mission requirements dominate the design constraints for operational architectures and thus drive the underlying system architecture and its technical foundation. However, mission does not define the entire environment. Other considerations include:

- Security
- Usability
- Robustness
- Sustainability
- Cohesiveness
- Economy
- Feasibility
- Coherence
- Affordability
- Modularity
- Connectivity
- Extensibility

6.1 Mission Focused

Critical mission requirements dominate development and selection of architectures. Data flows between domains must satisfy critical operational requirements driven by the **mission,** the **users**, and their **organizations**. "*The ultimate reason why a system is to exist is the filter through which all design decisions must successfully pass.*"[7] In the initial stages of analysis, comprehensive assessments[8] determine critical mission needs for users within diverse organizations. Production of a conceptual model of the mission space can augment this assessment and provide more detail. Without a comprehensive mission analysis, a review of existent systems and common data protocols can produce a quick estimate of critical data flows. However, reverse engineering breaks if missions or organizations radically change, if new technologies alter established data flows, or if operational architectures radically shift to adapt to threats or other stimulus.

6.2 Secure

Security defines the ability to tailor access to system components in concert with policy to enable forces to meet mission and/or training requirements. Security structures within the architecture should facilitate protection of data from intentional harm, misuse, or compromise. Concurrently they protect against natural disasters and accidental loss. Good security features facilitate rather than impede authorized access to information. In addition, they also establish dynamic methods to monitor and record transactions across the networks. Data records capture both authorized uses (baseline data) and illicit attempts to gain improper access to network segments or data structures. Good features also include automated methods to notify security personnel when unusual activity occurs or when users attempt to access or modify critical data. Security begins with focused assessments of vulnerability and risk for each specific use case. But, it is dependent on features constructed within the architecture that facilitate data redundancy, data partitioning, program layering, network segmentation, data filtering, data encryption, network traffic analysis and control, network gateways or firewalls, intrusion detection and response, and disaster recovery. [9]

Heretofore, security has been largely ignored in linking M&S and C4ISR domains. Our focus has been on enabling connections not establishing access security. As more systems are linked and more users added, the importance of security grows. In the future, security will be a major consideration in designing links between M&S and C2ISR domains. Training objectives and political goals grow demands to link federations to multi-national forces while each coalition force uses its native C2 systems. Operational, system, and technical architectures must support security features to permit authorized access while enhancing protection.

[6] The bibliography provides a non-exhaustive list of both directives and handbooks.

[7] Andriole, Information System Design, p48

[8] Production of the Universal Joint Task Lists (UJTL) provides generalized mission sets. However, it does not include a metric for data flows between organizations or staff elements nor does it develop supporting operational architectures.

[9] Sun Microsystems, Securing Networked Environments, Rev A, July 1997.

6.3 Feasible

Feasibility encompasses the state of the possible. In combination with mission needs and security, it completes the trio of the absolutes for potential architectures. While mission focuses on getting the right information to users to allow them to perform their tasks and security controls access to data and network segments, feasibility determines if the architecture is viable. In short, can all necessary components and connections be built? In addition, will the structure support requisite capacities?

The remaining ten considerations are subservient to the first three, but they are still extremely important to the success and long-term user acceptance of the final architectures.

6.4 Usable

Usability metrics extend beyond the bare-bones functions captured by mission needs or feasibility measures and reaches into "ease of use" and latency. Ease of use issues include user training; user interface devices; re-use of well-accepted components, and system configuration for unique user or mission requirements. Latency measures overall system performance and its responsiveness to user demands within mission parameters and resource constraints. Usability also balances automation and manual controls and proper distribution of tasks and allocation of control. However, all of these aspects of usability are tied back to basic requirements and mission needs.

> *"Unless we measure the extent to which the system satisfies user requirements and is compatible with the organization it is intended to support, then we really know very little about how good the thing is. Unfortunately, there is no correlation between the integrity of system algorithms and its ability to enhance human performance."*[10]

6.5 Coherent

Coherency combines the drive for consistent authoritative data, coordinated information flows, and a synchronized, unified data schema. This does not imply that perfect information will always be dispatched from the M&S applications to all segments of the C^4ISR system. Coordination and synchronization require that designated links provide consistent information in a temporal and geospatial context. At times, simulations may feed data developed with a specific perception that may not equate to perfect "ground truth". Coherency simply requires that the correct perception be fed consistently to establish the proper environment at the receiving segment of the target C^4ISR system.

Why is coherency important?

Uncoordinated data flows, conflicting data from multiple sources, and non-standard data formats degrade the utility of linking M&S and C^4ISR systems.

In today's environment, simulations produce abundant data. Unfortunately, extraction and correlation of seemingly incoherent data from multiple simulation sources or data feeds is a major burden borne by support teams or response cells in the information development and integration layer (Figure 1).

Coherent data stems from the extraction of the correct information from authoritative sources, processing the datum with verifiable and proven techniques, and delivering the information via a synchronized schema. Considerations include:

- Data ownership
- Data flows
- Metadata (data about the data)
- Data formats
- Data development algorithms

In tomorrow's environment, delivery of coherent data may be enhanced by M&S designs or by developing modules within C^2 systems to support information flows. However, neither can assure coherence if the connecting links and underlying architectures do not provide a synchronized delivery schema.

6.6 Robust

Robustness considers the amount and scope of external change or internal component failure that the architecture can accommodate. The dynamic challenge presented by shifting states of nature can be mastered by a combination of two approaches. First, flexible architectures can absorb changes and component failures without the need to shift the basic structure or to add unforeseen components. Second, adaptable architectures can quickly change to meet changing conditions in a timely fashion. In either case, the architecture must sustain external interfaces and support requisite data flows. Flexible architectures tend to be larger and support a wide range of known or expected situations. In contrast, adaptive ones are generally less extensive but quickly mutate to meet unforeseen circumstances.

[10] Andriole, Information System Design, p39

Regardless of the intended approach, the drive for robustness forces architects to design for changing conditions knowing that change is normal. Over time, system components wear out or degrade. Mission requirements evolve. New technologies emerge. Enabling technologies spread. Workload allocation strategies shift. Moreover, COTS and GOTS software follow separate development tracks. Hence, robustness – the ability to handle change – provides critical capability and fosters affordability.

6.7 Affordable

Affordability measures the cost to install, operate, and maintain (IOM) the system architecture that implements the technical architecture and enables the operational architecture. These costs include organizational, political, financial, training, and/or opportunity expenses. Affordability also encompasses other constraints placed on the design or development process and includes overall timeliness [11] in the delivery of potential solutions. It may compel the use of off-the-shelf software (COTS or GOTS) or re-use of other systems. It may drive prespecification of hardware and/or software components or limit design options for user interface modules. In addition, affordability may limit introduction of new technologies and thus dampen system performance or precipitate a reallocation of tasks to limit costs.

6.8 Sustainable

Sustainability encompasses a handful of subordinate factors. Most of these factors focus on the system architecture, but some overlap into the operational or technical architectures.

- Reliable – All system components complete essential tasks to established standards of performance over the requisite timeframe.
- Available – The system and its components are fully operational and accessible.
- Maintainable – System and its critical components are well designed and well documented to support repairs, upgrades, and migration to new platforms and/or operating systems.
- Observable – The actions and performance of the system and its components are open to inspection, assessment, and re-evaluation.
- Verifiable – The system and its components perform actions that are credible and traceable to the underlying mission requirements.

[11] Simmons, Software Measurement, pp192

6.9 Modular

Modularity captures an ability to repeatedly decompose the network and its components into smaller and smaller sub-systems until the parts are intellectually and technically manageable as independent units. Architectures may use spatial, functional, temporal, and/or logical modularity. "A modular system is composed of well-defined, conceptually simple, and independent parts interacting through well-defined interfaces."[12]

Common modular designs include client-server and/or subnet architectures. Advantages of these modular schemes include:

- Easier to understand individual modules
- Easier to understand relationships between modules
- Easier to document
- Easier to build
- Easier to test and integrate
- Easier to maintain

6.10 Cohesive

Cohesion marks the strength of the bonds among subordinate elements within a module. Modules with strong cohesion perform better. Moreover, implementations with highly cohesive modules generally have lower fault rates and lower costs than those with less cohesive modules.[13] From levels of high cohesion to levels of lower cohesion, a common scheme progresses through six layers.

- Data-type cohesion
- Object structure cohesion
- Functional cohesion
- Logical cohesion
- Sequential cohesion
- Incidental cohesion

6.11 Connected

Connectivity encompasses both physical and logical links between modules. Physical connections enable electromagnetic transmission of data streams. Logical links establish the protocols for data exchanges.

Physical bonds can employ a range of methods that are established by the operational requirements and the system constraints. For example, links can be established as point-to-point connections or network backbone hookups. In general, the physical

[12] Frakes, Software Engineering, pp27-28.
[13] ibid. pp28-29.

connections determine if data flows are unidirectional (simplex), bi-directional (duplex), or multi-directional (multi-cast or broadcast). Selection of physical options balances reliability, robustness, maintainability, economy, efficiency, and reuse.

In contrast to the physical links, coupling measures the strength of the logical linkages between modules based on information exchanges between them. In general, loosely coupled modules are better; they are easier to understand, document, code, test, and maintain.[14] One usable scheme defines the degree of coupling over range of six levels (from loosely to tightly coupled).

- Data definition coupling
- Data element coupling
- Object coupling
- Control coupling
- Global coupling
- Content coupling

Most current links implement loose coupling at the data element level with information passed through standard interface protocols. HLA links embody both data element and data definition coupling. In contrast, the DIS standard relied on object-level coupling. Most DII COE mission applications exhibit data definition coupling, but SHADE applications could inherit more tightly coupled schemes. Hence, SHADE modules can range from simple object coupling to control coupling to global coupling or even content coupling depending on specific methods used to make database exchanges.

6.12 Economical

Economy complements feasibility. It extends the feasibility concepts beyond determining if the architecture has the necessary capacity. It embraces the quest for the most effective, most efficient structure with sufficient capacity to meet mission and/or training requirements and considers the contributions made by all usable attributes of the system. Finding the optimum economical architecture does not equate to finding the cheapest. It balances performance against costs. It values usable features up to, but not beyond, their level of utility. In summary, it seeks the most effective operational value.

6.13 Extensible

Extensibility captures the level of difficulty encountered when extending the architecture to add capacity, clients, and/or functions. It ties into the potential for reuse of system components and builds upon adaptive attributes. Considerations include:

- Implementation of network topologies and communication protocols
- Application of open system standards
- Adherence to interface standards
- Adoption of programming standards
- Extent and accuracy of documentation
- Quality and scope of training materials and technical support

These thirteen design attributes form a basis for development and evaluation of potential architectures. The next section shifts the focus to an exemplary application of these factors. Byproducts include a crude evaluation of one prototype architecture. A definitive evaluation of this prototype is properly deferred to an interdisciplinary team of users, developers, and experts.

7. Application of Design Attributes

An architecture that overcomes common constraints employs a persistent domain data server (DDS) as the gateway to the C^2 domain. This gateway collects and holds data in a central location after it has been transmitted over the HLA RTI from the M&S domain.

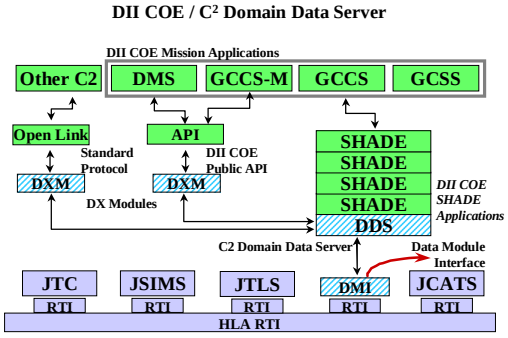

Figure 3: HLA RTI with Domain Data Server

This structure defines a demarcation line for developers on either side of the domain data server. Each can program an interface to well-known data protocols to insert datum into or extract information from the domain data server. This physically connected but loosely coupled structure encourages development of reusable modular components and provides structures to support implementation of security firewalls and gateways to network segments.

A coherent C^2 domain data server can promote reuse of common services and links to DII COE mission

[14] ibid. pp29-30.

applications. Over time, the best of these links may be segmented into the DII COE as standard services.

7.1 Potential Strengths

- Mission, user, organization requirements — Provides structure for data integration and coherent structure for data development for Joint training exercises.
- Security — Provides structure to support development of gateways, firewalls, network segmentation, and network monitoring.
- Usability — Standard connections mitigate training and support burdens.
- Coherence — Structure provides capability to ensure data coherence prior to entry into target C^2 systems.
- Robustness— Standard interfaces promote reusable links that improve flexibility, adaptability, and reuse. The data server's links to client applications within the C^2 domain are not constrained to use the HLA RTI. This flexibility opens the door for new topologies and adaptive structures.
- Modularity — Reduces maintenance. Supports client-server structures and network segments.
- Cohesiveness —Favors cohesive modules focused on single, reusable services.
- Connectivity — Loosely coupled via the HLA RTI to M&S. Domain data server is coupled via data protocols to DII COE mission applications and other C^2 systems. DDS may increase some development costs but should reduce operating costs. Design promotes reuse of links.
- Economy — Higher costs for development and maintenance of domain data server can be offset by reduction in operating costs and replacement of manual data integration with automated tools.
- Extensibility — Flexible and extensible. Links between C^2 and M&S leverage reusable and extendable protocols and pathways and can implement unique point-to-point links from between the domain data server and DII COE mission applications or other C^2 systems.

7.2 Neutral Attributes

- Feasibility — Although the technology isn't new, DoD C^2 data standards are still not completely defined and a domain data server has not been constructed. If these obstacles are overcome, remaining aspects of the architecture are low risk and mitigate other development obstacles.
- Sustainability— Potential for reduced software maintenance burdens. Best guarantee for production and delivery of reliable, available, observable, and verifiable data. However, production and maintenance of the C^2 domain data server may add expense and development effort.

7.3 Potential Weaknesses

- Affordability — Building and maintaining the C^2 domain data server adds development effort and expense.

8. Implications

Links between C2 and M&S domains are critical, but they should focus on obtainable and definable levels of interoperability. Links should leverage and adhere to established standards and specifications (if possible). Proposed design factors allow users to differentiate between architectures. Using a mission-based focus versus a technology-led approach, critical differences in architectures are captured in terms that relate to users, organizations, and missions. In addition, implications for architectures within the context of the DII COE emerge as a byproduct

8.1 Mission Focus

Links between domains must establish and maintain a focus on mission requirements. These links must facilitate both training and operational environments, and they must support all critical mission tasks. Information exchanges must provide mission-critical data in an integrated, coherent environment. In short, they must support an integrated operational architecture. In addition, solutions must be secure and sufficiently robust to handle dynamic shifts – not just evolution – in the operational architectures.

8.2 Secure

Security is fundamental to all levels of the architecture: operational, system, and technical. Connections between C^4ISR and M&S domains will expose underlying requirements to support multiple levels of security and multi-level security. In this context, multiple levels of security and multi-level security extend beyond the formal divisions of the DoD security classifications (i.e., unclassified, confidential, secret, top secret, and/or special access). It also includes subdivisions of users based on permissible time and type of access to data. However, these distinctions rely on supporting information. The operational architecture must provide time-dependent, dynamic user groups tied to their specific data requirements. In addition, metadata must identify attributes that allow differentiation of data items at the requisite level of

fidelity. In turn, secure technical and system architectures implement schemes to protect data integrity and to responsively deliver the right data – and only that data – to each authorized user.

Within the operational architecture, security stems from the ability to identify user groups that must exchange or access common data. These activities include groups authorized to read data, write data, and/or monitor metadata. Secure designs exploit operational modularity to limit information flow to a set of common, authorized users. In addition, catastrophic failures and inadvertent errors must be balanced by redundancy and data verification cells within the operational architecture. Status monitoring and response options must also be incorporated into the operational architecture. (In general, these features may be incorporated in the design of system architectures but are rarely considered in the development of the operational schemes.)

At the implementation level, viable security hinges on aligning system-architecture segments with operational-architecture domains. Each virtual domain should be designed to align with one level of security access. However, a given level of access may map multiple physical segments to its virtual domain. Likewise, a common physical segment may service more than one virtual domain by employing switches, switching routers, encryption techniques, and/or software filters. These (or other) methods are employed to establish boundaries between domains to create virtual networks of common users arrayed by their authorized access.

Within the technical architecture, security rests on inclusion of structures that facilitate access control, modular segmentation, and data coherency. Gateway and firewall structures between cohesive interface modules facilitate these goals. In turn, client-server structures facilitate reliable implementation of robust security structures and enhance coherent data flows across these boundaries. Given these goals, several design options are feasible.

8.3 Feasible and Robust

The logical and physical gateways between M&S and C^2 domains must be both feasible and robust. Robustness requires, as a minimum, that the architectures be both extensible and reusable. Moreover, they should be adaptable and/or flexible. In this context, flexibility and adaptability must permit the structure to meet dynamic requirements without reengineering basic components. Likewise, extensibility and reusability imply modularity and adherence to standards. Both of these attributes also contribute to the architecture's long-term viability.

8.4 Sustainable

The long-term viability of any proposed architecture hinges on its sustainability. The overall health of each interface module depends on its connections into both the C^2 domain and the M&S domain. Failure in either domain causes a connection failure.

Over the program lifecycle, loosely coupled connections using standard protocols are easier to maintain in comparison to unique point-to-point connections. For the M&S domain, this approach translates into stipulating reuse of a common Simulation Object Model (SOM) for links with C^2 systems.

At the other end of the bridge, the C^2 environment hinges on the DII COE and compliant mission applications. Hence, long-term sustainability of interface modules would be enhanced by their integration into the COE. Thus, all components on the C^2 side of the HLA RTI should be considered for migration into the DII COE as M&S support modules. In addition, system architects should specifically consider integration of a coherent domain data server into the DII COE SHADE.

Integration of M&S modules into the DII COE implies movement toward common data structures and data definitions. Over time, use of common terms and structures can evolve into a comprehensive, enterprise-level, data model encompassing M&S, C^2, and other information system domains. In the near term, simple, common structures will enable the production of reusable data-exchange modules. These modules leverage a collection of base-object models (BOM) to map C^2 data items to objects and attributes within the simulation domain. Collectively, a set of base-object models combine to form a hierarchical-object model (HOM) that fuses multiple BOM to produce data streams in standard formats and protocols. In terms of the proposed Simulation Interoperability Standards Organization (SISO) definition, the data items that feed a comprehensive set of hierarchical-object models for C^2 interfaces could provide the template for a reference federation object model (RFOM). Data model reuse couples with standardization to enhance sustainability.

9. Conclusion.

In conclusion, the specified factors allow users to make meaningful operational distinctions between technical

and system architectures. Good designs can enhance security and mitigate overhead workloads in response cells in training exercises. Moreover, a mission-based focus also implies closer ties to the DII COE (for Joint forces) and emphasizes the need for standardization of data between M&S and C^2 domains.

10. References

[1] Andriole, S. J., Information System Design Principles for the 90's—Getting It Right! 1990, Fairfax, VA, AFCEA International Press

[2] Carpenter, H., 11 July 1997, "Methodology for Assessment of C4I Architectures," MITRE Technical Report No. W150-M-083, The MITRE Corporation, Reston, VA

[3] Chairman Joint Chiefs of Staff Instruction, *Global Command and Control System Configuration Management Policy*, CJCSI 6722.01, 1 July 1997, Joint Staff, Washington, D.C.

[4] Chairman Joint Chiefs of Staff Instruction, *Joint Modeling and Simulation Management*, CJCSI 8510.01, 22 December 1995, Joint Staff, Washington, D.C.

[5] Chairman Joint Chiefs of Staff Instruction, *Joint Strategic Planning System*, CJCSI 3100.01, 1 September 1997, Joint Staff, Washington, D.C.

[6] Chairman Joint Chiefs of Staff Instruction, *Procedures for Compatibility, Interoperability, and Integration of C3I Systems*, No. 4630.8, Assistant Secretary of Defense, DOD, Washington, D.C.

[7] Chairman Joint Chiefs of Staff Instruction, *Requirements Generation System* (Formerly MOP 77), CJCSI 3170.01, 13 June 1997, Joint Staff, Washington, D.C.

[8] Chairman Joint Chiefs of Staff Instruction, *Verification, Validation, and Accreditation of Joint Models and Simulations*, JSI 8104.01, 12 January 1995, Joint Staff, Washington, D.C.

[9] Chairman Joint Chiefs of Staff Manual, *Employing Joint Tactical Communications—Joint Network Management and Control*, CJCSM 6231.07A, 24 January 1997, Joint Staff, Washington, D.C.

[10] Chairman Joint Chiefs of Staff Manual, *Joint Training for the Armed Forces of the United States*, CJCSM 3500.03, 1 June 1996, Joint Staff, Washington, D.C.

[11] Chairman Joint Chiefs of Staff Manual, *Universal Joint Task List*, Version 3.0, CJCSM 3500.04A, 13 September 1996, Joint Staff, Washington, D.C.

[12] Department of Defense Directive, *Compatibility, Interoperability, and Integration of C3I Systems*, No. 4630.5, 12 November 1992, Assistant Secretary of Defense, DOD, Washington, D.C.

[13] Department of Defense Directive, *Defense Information Management (IM) Program*, No. 8000.1, 27 October 1992, Department of Defense, Washington, D.C.

[14] Department of Defense Directive, *DoD Data Administration*, No. 8320.1, 26 September 1991, Department of Defense, Washington, D.C.

[15] Department of Defense Directive, *Management and Control of Information Requirements*, No. 8910.1, 11 June 1993, Department of Defense, Washington, D.C.

[16] Ferraby, L., 1991, Change Control, Hertferdshire, UK, Prentice-Hall International (UK) Ltd.

[17] Frakes, W. B., and C. J. Fox, B. A. Nejmeh, 1991, Software Engineering in the Unix/C Environment, Englewood Cliffs, NJ, Prentice-Hall, Inc.

[18] JIEO/JITC Circular, *Requirements Assessment and Interoperability Certification of C4I and AIS Equipment and Systems*, JIEO/JITC Circular 9002, 23 January 1995, Defense Information Systems Agency, DOD, Washington, D.C.

[19] Knoefel, J. O., July 1992, *C4I For The Warrior*, Briefing by Naval Electronic Systems Engineering Activity, NESA Code 2400, St. Inigoes, MD

[20] Roetzheim, W. H., 1991, Developing Software to Government Standards, Englewood Cliffs, NJ, Prentice-Hall, Inc.

[21] Simmons, D. B., and N. C. Ellis, H. Fujihara, W. Kuo, 1998, Software Measurement: A Visualization Toolkit for Project Control and Process Improvement, Upper Saddle River, NJ, Prentice-Hall, Inc.

[22] Theater Battle Management Design Team, The MITRE Corporation, July 1992, *Technical Approach to the Future Theater Battle Management C4I Architecture for Deployable Operations*, Briefing No. M92B0000073, Electronic Systems Center, Hanscom AFB, MA

[23] Thompson, B., The MITRE Corporation, 23 June 1997, *Levels of Information Systems Interoperability (LISI)*, Briefing for Program Sponsor: C4I ISA Architectures Directorate

Author Biography

KEVIN BRANDT is a lead engineer for modeling and simulation for MITRE at USACOM Joint Warfighting Center. His focus centers on user requirements and development of models for Joint training simulations. He has designed and executed Joint and coalition training exercises and demonstrations with data feeds between training simulations and C2 systems. He is currently developing architectures for the Distributed Joint Training (DJT) program.