Initialize. Train. Initialize. Win.

Dr. Amy E. Henninger
Institute for Defense Analyses (IDA)
Alexandria, VA
ahenning@ida.org

Dr. William J. Gerber
IDA Consultant,
WJ Gerber Consulting
Orlando, FL
bill.gerber@ieee.org

Mr. Paul Works, Jr.
Methods and Research
TRADOC Analysis Center
Ft. Leavenworth, KS
paul.works@us.army.mil

Dr. Michael Hieb
Center of Excellence in
George Mason University
Fairfax, VA
mhieb@gmu.edu

ABSTRACT

As a result of various digitization initiatives, Army Battle Command (BC) systems have evolved into sets of interconnected systems, forming synchronized information architectures. Operationally, the purpose of these information architectures is to establish and maintain a distributed, consistent understanding upon which organizations execute synchronized operations. Underpinning this capability for distributed understanding is the need for distributed, consistent data. Each BC system must be initialized with that consistent data to be able to synchronize with the other BC systems. And, to support the requirements to “train as we fight”, modeling and simulation (M&S) systems must also be able to synchronize with these BC systems, using that same consistent data. This is an enormous challenge.

Currently, there are multiple initialization processes executed by multiple organizations using multiple tool sets for multiple systems (e.g., modeling and simulation, battle command, and communications networks). The cost in time and resources to initialize all of these systems is perceived to be excessive, and the full range of Army systems and processes that perform initialization is not well understood much less streamlined. As the Army moves towards digitization and as embedded, inter-vehicle training systems become a reality, the inefficiencies and overlap in these processes become a costly impediment. Rapid, repeatable and error-reducing initialization processes and tools to implement those processes must be available to both the BC and M&S systems.

Sponsored by SIMCI (PEO STRI and PEO C3T), this paper will present the analysis of initialization requirements for BC and communications systems and M&S systems used by a Heavy Brigade Combat Team (HBCT). It will detail the methodology used to collect data and present the results to include: a characterization of the common data for BC, communications, and M&S; an estimate of resources required to derive these data; and recommendations for future work.

ABOUT THE AUTHORS

Dr. Amy E. Henninger is a member of the Research Staff at the Institute for Defense Analyses (IDA) where she provides technical guidance and analytic support to a variety of Modeling and Simulation (M&S) programs across the Department. A former I/ITSEC Scholarship Recipient and a former Adjunct Faculty Member for the M&S Graduate Program at the University of Central Florida, Dr. Henninger has an extensive publication record and holds six degrees including a Ph.D. in Computer Engineering.

Dr. William J. (Bill) Gerber, Lt Col, USAF (Ret.), is Principal Researcher for WJ Gerber Consulting and consults on various projects for the Institute for Defense Analyses (IDA). He is also a Staff Technical Analyst for Dynamics Research Corporation (DRC), where he primarily works as a member of the staff of the US Army’s Future Combat Systems (FCS) Lead Systems Integrator (LSI) Training Integrated Product Team (Training IPT). A former I/ITSEC Scholarship Recipient and Link Fellow, Bill has a B.S.E.S. from the US Air Force Academy (USAFA), an M.S.E. from the University of California at Los Angeles (UCLA), and an M.S.Cp.E. and Ph.D. from the University of Central Florida (UCF). He is a Certified Modeling and Simulation Professional (CMSP).

Mr. Paul Works, Jr. is the Chief, Methods and Research Office, TRADOC Analysis Center (TRAC), Ft. Leavenworth, KS. He advises Director, TRAC on matters pertaining to advanced methodologies, technologies, and modeling and simulation (M&S) tools. Mr. Works has worked as an Operations Research and Systems Analyst (ORSA) for TRAC since 1998, and previously as a physicist/ORSA for the Weapons and Tactics Analysis Center (WEPTAC) and the Advanced Systems IPT at the Naval Air Warfare Center - Weapons Division (NAWC-WD), China Lake, CA. Mr. Works has a B.S. in Physics and Mathematics and a M.S. in Operations Research from Kansas State University.

Dr. Michael Hieb is a Research Associate Professor with the Center of Excellence in C4I at George Mason University. Dr. Hieb was the Co-Chair of the SISO CBML Study Group and also was on the team that developed the initial BML concept for the US Army. He received his PhD in Information Technology at George Mason University in 1996, developing an instructable Modular Semi-Automated Forces agent. He has published over 90 papers in the areas of Formal Languages for Command and Control, Simulation Interoperability, and Multistrategy Learning.

Initialize. Train. Initialize. Win.

Dr. Amy E. Henninger
Institute for Defense Analyses (IDA)
Alexandria, VA
ahenning@ida.org

Dr. William J. Gerber
IDA Consultant,
WJ Gerber Consulting
Orlando, FL
bill.gerber@ieee.org

Mr. Paul Works, Jr.
Methods and Research
TRADOC Analysis Center
Ft. Leavenworth, KS
paul.works@us.army.mil

Dr. Michael Hieb
Center of Excellence in
George Mason University
Fairfax, VA
mhieb@gmu.edu

1.0 INTRODUCTION

Initialize. Train. Initialize. Win. What does initialization have to do with training and winning? Nowadays, the answer is a whole lot.

High-intensity operations during Operation Enduring Freedom (OEF) and Operation Iraqi Freedom (OIF) have called attention to the initialization problem. For example, during the OIF assault on Baghdad, a number of Army Battle Command System (ABCS) systems became useless in battle because the systems crashed and could not be reinitialized on the move. The 20-minute halt required for reinitialization from the network was not an option while in an attack (Sprinkle and Brown, 2004). In another example, during 4ID Operational Evaluation in April 2005, a mismatch in Uniform Resource Name (URN) assignments between products crippled the ability to do unit task reorganization (UTR). When the soldier tried to change the association of one unit, the URN of another unit changed instead (Blalock, 2005).

What about modeling and simulation (M&S)? During operation Millennium Challenge, duplication of IP addresses caused numerous messaging applications to fail on a significant scale. In another example, during ABCS 6.4 testing, the mismatch of identifiers caused feeds from the Common Operational Picture (COP) to come across on ABCS and Force XXI Battle Command Brigade and Below (FBCB2) displays as unknowns (Sprinkle and Black, 2006).

There are numerous documented errors resulting from systems having erroneous data at initialization. Everything from "ghosting" platforms to failed Domain Name Server (DNS) servers, failed messages, unknown icons, network failure, and wrong units/systems being referenced can occur as a result of bad data.

1.1 Initialization Defined

There are various definitions of *initialize* or *initialization*. The following definitions provide some background.

In the classic introductory programming book "Oh! Pascal!", Cooper and Clancy (1982) describe the concept of initialization as *"A variable must be initialized, or given a starting value, before it can appear in an expression. The variable is undefined until it is initialized."* Here note that variables can't be used until initialized.

From Dictionary of Computing (Collin, 2002), initialize is a verb meaning *"to set values or parameters or control lines to their initial values, to allow a program or process to be re-started."*

Many general purpose simulation languages (e.g., SLAM and GPSS) supply control statements to assist with initialization of a host of different simulation-specific data (e.g., beginning time of run, finishing time of run, clearing of statistical arrays, and program variable initialization). Because a simulation is basically an executable database, there are numerous values that require initialization prior to runtime and many of these are unique to the individual simulation.

Distributed M&S systems are significantly more complex to initialize (Prochnow et al, 2005). Not only is there a unique set of initialization data for each of the components, but there is also a common set that enables the components to communicate (i.e., meaningfully exchange data). Derivation of this common initialization data is complicated by the fact that these components are heterogeneous systems that do not share a common software engineering baseline. Thus, there are many system-specific issues (e.g., naming conventions, dictionary of existing entities, and data formats required) that add to this complexity.

As evidenced in the examples in the beginning of the introduction, distributed M&S, however, is not the only domain to experience this challenge. For example, in the battle command (BC) domain, systems such as the ABCS require data to initialize the variety of components that comprise that system, and some of these data are unique to those components and other data are common across them all. Also, the Signal Corps, working with the domain of communications systems and networks, is faced with these same challenges. In order for the radio systems to work, they must be initialized along with the network.

If one considers all of these domains interacting, as in a very large Live/Virtual/Constructive (LVC) exercise, then he can begin to appreciate the magnitude of the challenge in initializing Systems of Systems. That is, as the number of systems increases linearly, the complexity of the initialization challenge increases non-linearly. Not only are there unique data that require definition, there are common data across all of the systems in these domains that require definition and it must be done in the context of systems that do not share common software baselines, as well as in the context of developers and operators who do not share common backgrounds (i.e., interpretation of initialization requirements becomes dependent on the background of the operator).

For purposes of this paper, initialization has been defined in broad terms and is considered to be information needed by models and simulations, communication systems, and battle command systems before they can be used in operations, training, testing, or experimentation. Truly, initialization can apply to other relevant information systems (e.g., sensors, weapon systems, and smart munitions), but since none were investigated for this paper, we do not include them in the scope of the definition used here.

1.2 Why Study Initialization?

BG Nickolas Justice's (PEO C3T) comments to a 2006 forum dedicated to investigating issues in initialization (Connors et al, 2006) included both "Initialization is the #1 challenge facing the Modular Force." and "The complexity of the Initialization problem is geometric in scale – and due to the complexity, most do not understand it."

Problems exist with the BC initialization processes and tools. First, the cost in time and resources to initialize current US Army BC systems is excessive. The timeline for the current US Army BC initialization process for a Heavy Brigade Combat Team (HBCT) is on the order of multiple months and the costs are excessive. Second, the full range of Army systems and processes that perform US Army BC system initialization is not well understood and there is no single and agreed-upon identification of these systems and processes.

As BC systems become more interdependent, training and testing systems also evolve to support them and their need for consistent data. M&S systems now play major roles in training and testing, requiring the same consistent data for initialization as the BC systems with which they interact. To avoid an impact to testing, training, and operations, rapid, repeatable and error-reducing initialization processes and tools to implement those processes must be available to both the BC and M&S systems.

A goal of developing the initialization process and tools is to automate the data input to achieve system or system of systems (SoS) startup conditions. Automation is essential to reduce initialization errors and time and to facilitate distribution. It is hard to imagine that the manual process (generating errors and taking weeks to load) could effectively support systems in combat. Figure 1 depicts data error examples introduced by manual input.

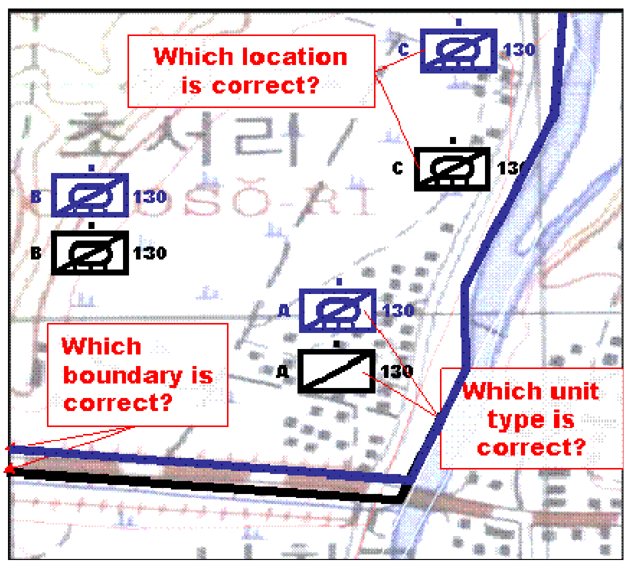

Figure 1. Example of AFATDS icons in Blue and FBCB2 icons in Black[1]

2.0 METHODOLOGY

Before embarking on any attempts to improve initialization processes, it is necessary to understand the processes currently in place for the various systems

[1] Recreated from Sprinkle and Black, 2006.

and how the processes for these different systems interact at the seams, to get the systems-of-systems perspective. Moreover, it is important to have some baseline metric established such that future enhancements can be measured. This paper, intended to be educational in nature, reviews some of the processes and establishes metrics to quantify efficiency of these processes.

To establish this baseline and subsequent metrics, the study team used an expert-based approach. Specifically, we interviewed a number of experts on a variety of different battle command, communications, and M&S systems; documented the findings; developed appropriate assumptions; synthesized the results; and then continued the latter part of this cycle to refine the estimates and verify the results.

Throughout the study, we make the observations that initialization data and processes can be categorized into:

- Interoperability data,
- Stove-Pipe (unique) data, and
- M&S specific data.

Further we hypothesize that the relationship between the categories of data is approximated by the representation in Figure 2 below.

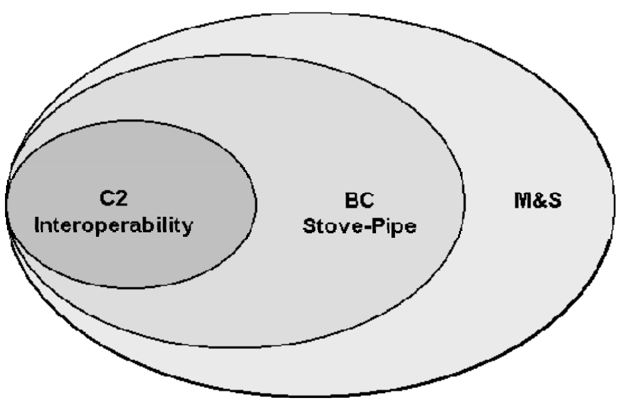

Figure 2. Nested Data Types

The following three sub-sections review the processes used to determine initialization requirements for battle command and communications systems, the M&S systems, and the interoperability data required across all of the systems.

2.1 Battle Command and C2 Data

This section presents a high-level overview of the Army Battle Command System (ABCS) and then a more detailed review of the individual components and their general data requirements for initialization. These requirements were mostly provided by a subject matter expert (SME) at the Central Technical Support Facility (CTSF) at Ft. Hood. Some were also generated from discussions at Program Director Mission Network Operations (PdM NetOps) at Ft. Monmouth, and finally some were mined from related documentation.

ABCS is a complex system of systems that link automation assets, communication media, and operational facilities (Greene and Mendoza, 2005). These systems are fed data from satellites, aerial reconnaissance, weapons systems, sensors, and ground soldiers; and ultimately provided to commanders. It provides them the ability to collect and analyze information, develop plans and orders; and then eventually disseminate this information both to lower echelons in the command chain by directing forces towards objectives and also to upper echelons where the tactical battlefield is monitored and coordination among joint forces is performed[2].

The ABCS consists of a number of battlefield automated systems, each supporting soldiers specializing in a battlefield functional area (BFA) including: Maneuver, Fire Support, Air Defense, Intelligence/Electronic Warfare, and Logistics. ABCS allows commanders to request, select, and evaluate data from diverse resources to create relevant information and maintain the common operational picture (COP) (Frambes, 2005; Moore, 2007). Table 1 displays the systems located in the Command Post (CP) and Figure 3 provides a sampling of the data types they contribute to the COP.

The data exchange networking format used by ABCS 6.4 is the Lightweight Directory Interchange Format (LDIF). LDIF is an American Standard Code for Information Interchange (ASCII) file format used to exchange data and enable the synchronization of that

[2] ABCS is the Army's tactical component of the Global Command and Control System (GCCS). The Theater Battle Management Control System (TBMCS) is the Air Force's tactical component. The Joint Maritime Command Information System (JMCIS) is the Navy's tactical component, and the Tactical Combat Operations System (TCO) is the Marine's tactical component of the Global Command and Control System.

Table 1. BFA Picture Contribution to COP

BFA	ABCS System
Maneuver	Maneuver Control System (MCS) The Force XXI Battle Command Brigade and Below (FBCB2)
Fire Support	Advanced Field Artillery Tactical Data System (AFATDS)
Air Defense	Air and Missile Defense Work Station (AMDWS)
Intelligence / Electronic Warfare	All-Source Analysis System (ASAS)
Logistics	Battle Command Sustainment Support System (BCS3)
Other Systems (not BFA specific)	The Global Command and Control System-Army (GCCS-A)
	Digital Topographic Support System (DTSS)
	Integrated Meteorological System (IMETS)
	Integrated System Control (ISYSCON)
	Tactical Airspace Integration System (TAIS)

data between Lightweight Directory Access Protocol (LDAP) servers. LDAP defines a directory access protocol mainly over the Transmission Control Protocol/Internet Protocol (TCP/IP) suite of protocols.

The LDIF is important in the networking process of ABCS 6.4, as the system administrator or G6 depends on the LDIF to configure the network within set roles; unit reference numbers (URNs), subnet masks and names, and addresses (gateway, subnet, and IP).

Proper configuration of the network ensures data exchange. Essentially, to facilitate data exchange, these data translate into four critical pieces of information for each system listed:

- Who are you? (role name)
- Who do you need to know? (address book requirements)
- Where are you? (network address)
- What version of software are you running? (compatibility/functionality)

For each of the components shown in Figure 3, we documented the major data types, the formats, the data source, the media type, and then a manpower estimate of how long it would take to load the data such that the component was ready for operations. Figure 4 shows this documentation for one of the ABCS components, Advanced Field Artillery Tactical Data System (AFATDS).

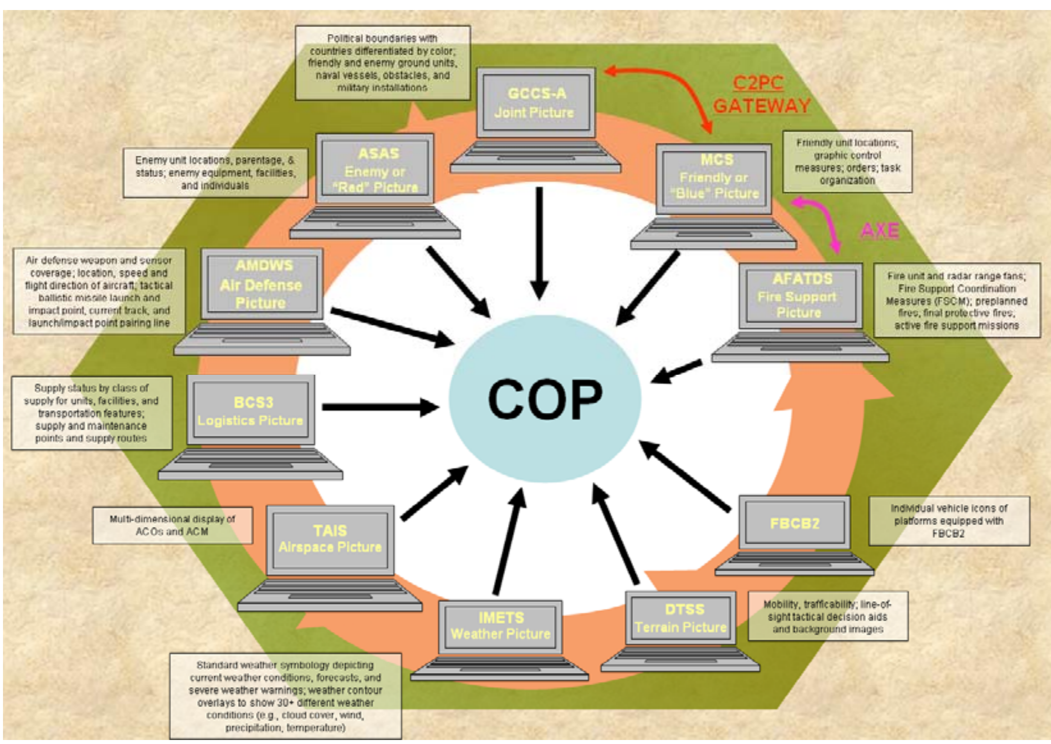

Figure 3. BFA Picture Contribution to COP with Data Type Samples

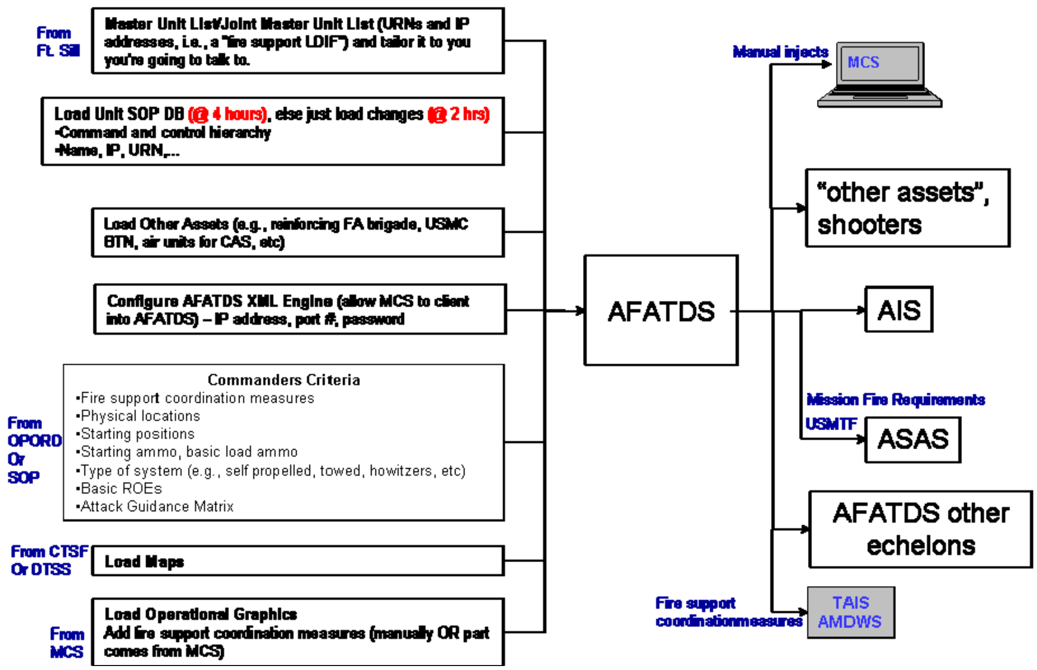

Figure 4. AFATDS Initialization Data Requirements

After developing an estimate of what is required to initialize the AFATDS in a generic instance (i.e., absence of operational context), we multiplied these resource requirements by the number of AFATDS in a Heavy Brigade Combat Team (HBCT) and that became a baseline number for the total resources required to initialize all AFATDS in an HBCT.

For the communications equipment, we repeated this process for radio sets. For a HBCT, this included over 17 types of radios[3] communicating voice or position, and each radio required assignments for Internet Protocol (IP) address, frequency pool, and COMSEC.

2.2 M&S Data

The range of simulation environments that could be coupled with the Battle Command and Communications systems identified in the previous section is vast. Integration of these three domains could occur in training exercises, testing events, experimentation events, or even operations and mission rehearsal. For context, the M&S wrapper investigated thus far is the Army's Joint Land Component Constructive Training Capability (JLCCTC) Entity Resolution Federation (ERF), an exercise driver designed to facilitate Battlestaff collective training for brigade or below levels. The ERF provides a simulated operational environment in which computer generated forces stimulate and respond to command and control processes that the personnel have in the field, and the training audience interfaces with the simulation environment via C2 devices, the same tools used to communicate in real world battlespaces.

Figure 5 illustrates the components of the JLCCTC ERF and categorizes them according to whether they're a simulation, C4I interface, AAR tool, federation tool, C4I system, or networking and communications infrastructure.

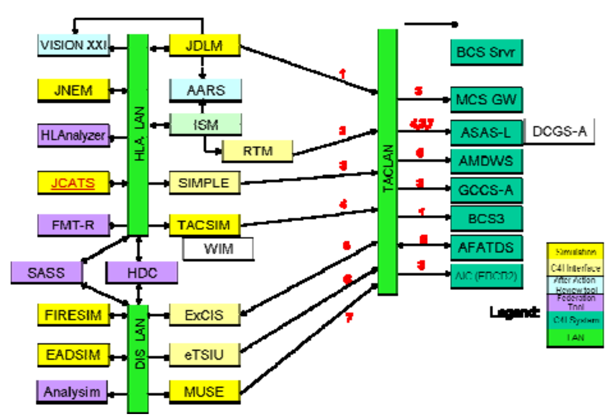

Figure 5. JLCCTC ERF Architecture

[3] DAGR, EPLRS, PEWS, PRC-117, PRC-119, PRC-126, PRC-150C, PSC-5, VRC-83, VRC-87F, VRC-88F, VRC-89F, VRC-90F, VRC-91F, VRC-92F, VRC 103 and VRC-104

The Joint Conflict & Tactical Simulation (JCATS) is the maneuver driver simulation for the federation, and as such, all other simulations and stimulators of the federation follow its lead. Most of the units reside in the JCATS database, and some of the units are in other federation members' databases exclusively, e.g., units in FIRESIM, LOGFED, and TACSIM. The database build process for ERF, shown in Figure 6, is a mix of manual, semi-automated, and automated processes. To develop initialization requirements (e.g., data types, formats, and providers; transforms applied; and resources required), we distributed detailed surveys on the various steps in the process (shown in Figure 6)[4].

2.3 Data Product Development: Interoperability Data

In Figure 2, we saw that a core set of interoperability data is required to support the Battle Command Systems and their integration with M&S. This interoperability data comes in the form of a "data product" (DP) developed by the Data Product Development Environment (DPDE), formerly known as ACSIS (Carlton and Scrudder, 2003).

This data product is a set of files in various formats that are needed by systems participating in some event that requires integration across these systems. For example, it provides C4ISR initialization data products for the units deploying to OEF and OIF, and some simulation initialization data products to support LVC exercises. The set of files created is dependent on the implemented architecture. ABCS 6.4, for example, requries an **L**DAP (Lightweight Directory Access Protocol) **D**ata **I**nterchange **F**ormat (LDIF) and Address Book for general use by all systems, and unique data loads are prepared for some ABCS systems, e.g., AMDWS, BCS3, etc. Data products for multiple architectures are maintained in the DPDE database with each being uniquely identified by an index and version number known as a Unit Task Organization (UTO).

As shown in Figure 7, DPDE takes approximately 12 weeks to define, de-conflict, and generate the initialization data products. This is accomplished only after the delivery of a unit systems architecture (SA). The construction of the SA is itself a laborious and lengthy process which takes between 12 – 24 weeks. However, once an initial unit task organization (UTO) is built by DPDE, variations of the UTO can be created in minutes. As a result of the DPDE process, the Army is vastly more confident in the final data product quality than had been previously possible.

At the end of the process, DPDE-supplied data includes Force Structure Data, Network Structure Data, Command and Control Data, and Entity Level Data. These are all described below.

2.3.1 Force Structure Data

Force structure data is a unit hierarchy as described by unit name, unit identification code (UIC), unit equipment and unit billets. It can include different sides (opposing, coalition, and neutral) and domains (ground, air and sea) force structure data.

2.3.2 Network Structure Data

Network structure data includes all of the information required to support network initialization: unit name, role names, URNs for all pieces of digital equipment (radios, routers, switches, battle command systems, etc), IP addresses, subnets, router configurations, multi-cast groups, and email addresses.

2.3.3 Command and Control Data

Command and control data is required to support integration of M&S applications into battle command systems. It includes all operations data related to plans and orders with accompanying overlays, matrices, and control measures.

2.3.4 Entity-Level Data

Generally, BC systems are only concerned with organizations and platforms that have BC related digital systems. They are not concerned, for example, with voice-only radio systems. Likewise, they are not interested in initialization for most weapons; nuclear, biological and chemical (NBC) equipment; individual warfighters (billets); organizations below platoon level; and the relationships of organizations to billets to equipment. Many simulation systems, on the other hand, are interested in entity-level data because it associates attributes and behaviors with organizations, platforms, and billets.

3.0 INITIAL RESULTS

Table 2 shows the roll-up of all the data discussed (battle command and communications, M&S, and inter- operability data provided by DPDE) in previous sub-sections. Given all of the assumptions, reported in

[4] These data are still being analyzed. Results reported in this paper represent an initial estimate.

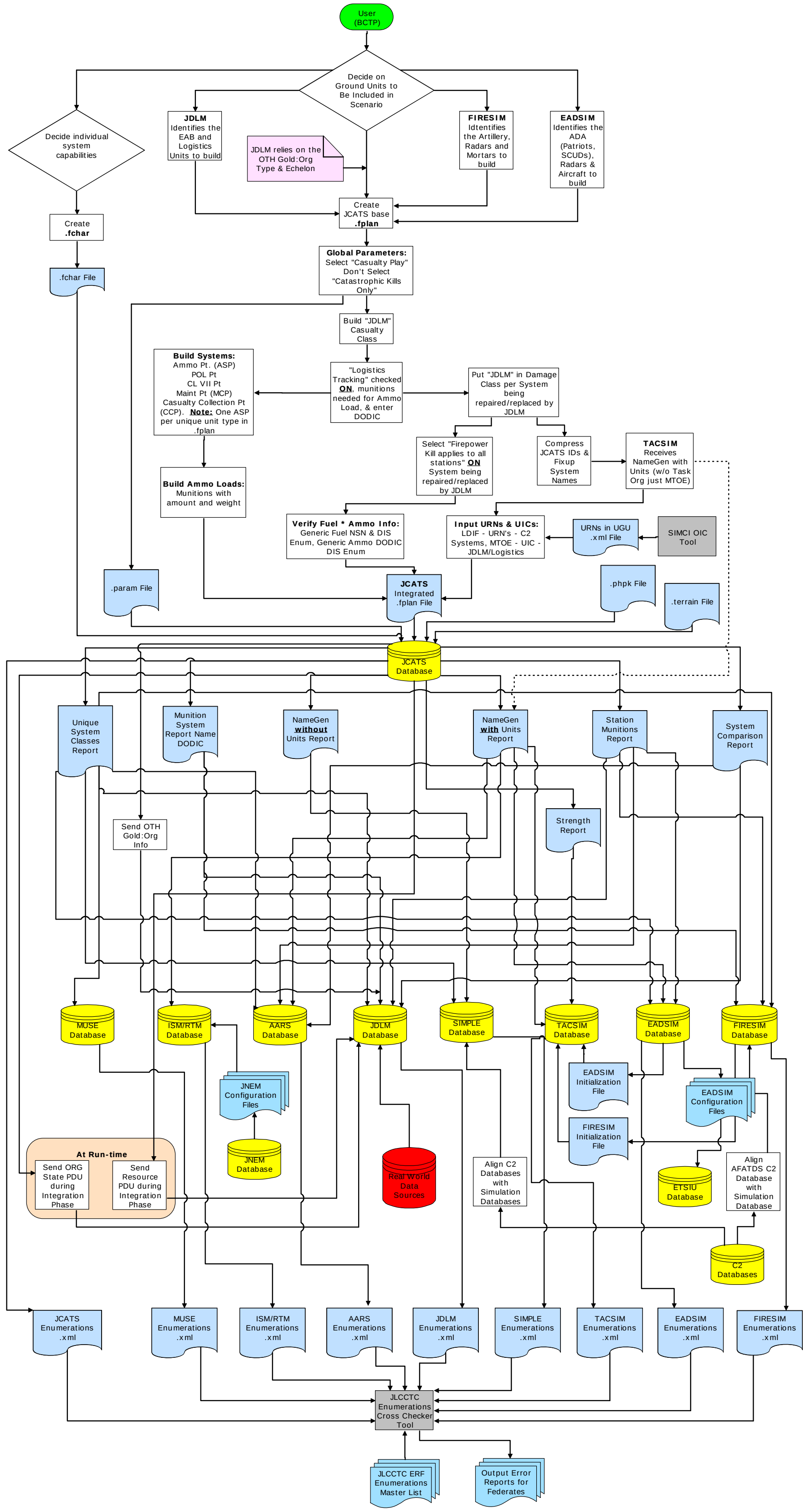

Figure 6. JLCCTC ERF Database Build Process

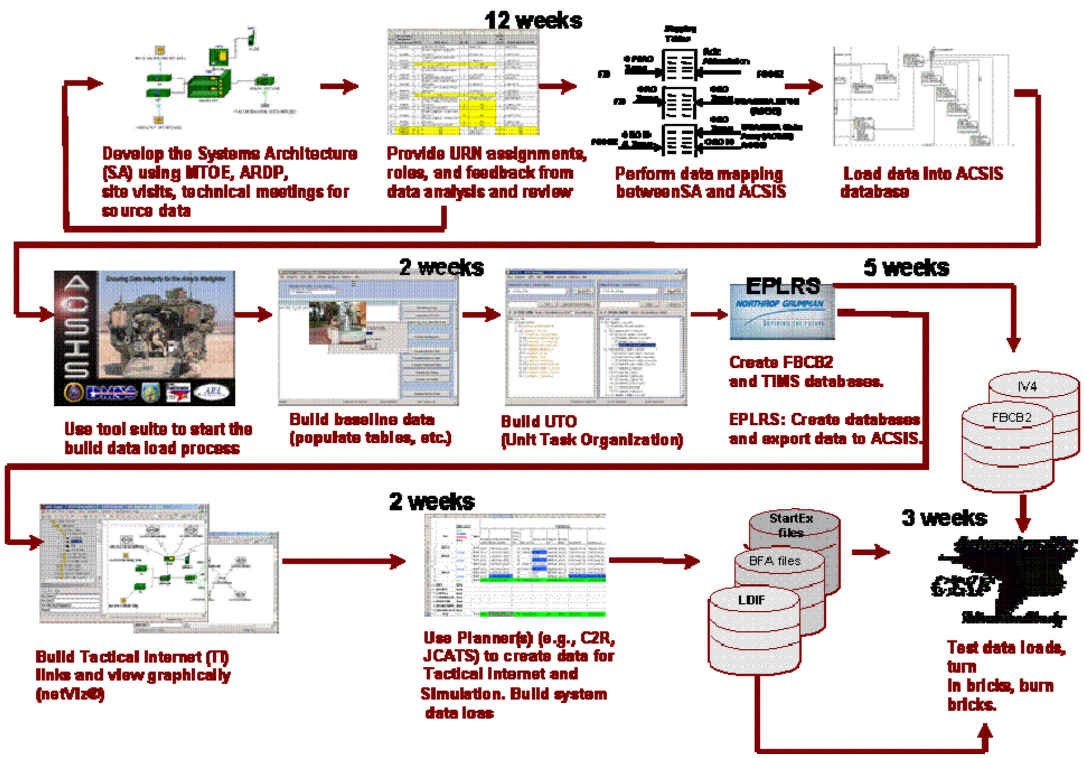

Figure 7. DPDE Data Product Build Process

Table 2. Preliminary Results

Process/Systems	Estimated Man-Hours
Data Product Development (Base Layer Only)	2576
M&S (Base Layer Only)	1124
Battle Command (Does Not Include OPORD graphics)	816
Radios (Base Layer Only)	857
Estimated Total Man-Hours =	5373

detail after the table, the preliminary results suggest that it takes over a man-year to develop the interoperability data for a HBCT and over a man-year to develop the stove-piped data for battle command and communications and M&S systems. In all, we estimate that it takes approximately 2.5 man-years to initialize all of the battle command and communications systems and M&S systems for a HBCT in a JLCCTC ERF-based exercise.

It's important to note, however, that this initial estimate has been developed with a number of assumptions. For example, all quantities of systems were based on the May '07 HBCT template from PDM NetOps. This template represents a superset of all components found in an HBCT. Likewise, the estimate does not yet include estimates on any of the underlying data products or processes. For example:

- the estimate of the Data Product development assumes the availability of a system architecture template (i.e., SA does not need to be developed from scratch),
- the estimate of the Data Product development includes only LDIF and FBCB2 Data Base data products,
- the estimate of M&S data assumes that AMSAA has already produced system specification data,
- the estimate of the Battle Command data assumes that OPLAN/OPORD graphics have been developed (e.g., on MCS) and can be imported into BC systems,
- the estimate of terrain data used to support BC and M&S assumes that the appropriate Terrain Data Bases (TDBs) are available,
- the estimate of Radio data assumes that crypto initialization has been completed and that spectrum allocations have already been developed,
- the estimate of Radio data assumes that everything is grounded, all antennas are up, etc., and
- the estimate does not include network initialization.

In addition to all of these caveats, the current preliminary estimates assume that the Operator has access to required information (e.g., the operating system, supporting software, and interoperability data products are easily accessible) and that the operator is well trained. Thus, we expect the entire process of initialization, cradle-to-grave, requires many more resources than what is reported in Table 2.

4.0 CONCLUSIONS

As reported in Carr et al (2007), the provision and management of modeling and simulation (M&S) data in support of the Army's various user communities has become increasingly problematic as M&S systems have expanded in scale and scope, and as more and more simulations are "federated" with others, and with companion C4ISR systems in the conduct of training exercises. In short, the axiom that "if the data are not interoperable, the systems are not interoperable" is proven again and again by long and difficult integration periods—often lasting months, or even years—in which simulation federations are built and tested until they are finally ready for their end users. In today's highly dynamic warfighting environment, such long lead times are increasingly unacceptable.

Coupled with this phenomenon is the increasing number and complexity of M&S systems in support of various communities—not only training, but acquisition, test and evaluation, analysis, and experimentation—and the increasingly broad variety of data they consume. For instance, environmental data may range from the very large-scale provision of geospatial data only (such as in support of a global- or theater-level simulation) to the extremely fine-grained geospatial, weather, and oceanographic data needed to support a flight simulation for a single aircraft. These issues clearly point to the need for efficiencies in creating data, and for interoperability among data providers and integrators.

Data management is the foundation of any system that has as one of its high priority requirements the reliable, and consistent exchange of mission critical information. Throughout the study, it was apparent to us that data manipulation is important, but it is doable. The more critical issue seemed to be acquiring data from an authoritative data source (ADS) and using that source rigorously across different data consumers and/or integrators. This observation especially rang true for both geospatial and C2/simulation interoperability data.

5.0 ACKNOWLEDGEMENTS

Many organizations are working to improve scenario generation and initialization processes for M&S. To our knowledge, none have had the fortitude to actually document processes to the level of detail shared in this paper. We believe that before a process can be improved, it must be understood and it must be measurable. More foundational work, like this, should be done before any large-scale investments are made in efforts to improve initialization and/or scenario generation. Without this kind of foundation, the improvements touted by large-scale efforts will not be measurable.

This work was performed under a SIMCI Study sponsored by the U.S. Army Program Executive Office for Simulation, Training and Instrumentation (PEO STRI) and U.S. Army Program Executive Office Command Control Communications Tactical (PEO C3T). That support is gratefully acknowledged.

6.0 REFERENCES

Blalock, J. (2005). Army Initialization Capability Strategic Approach White Paper, Chief Information Officer/G-6. Department of Army.

Carlton, B. and Scrudder, R. (2003). An Integrated Solution to C4I and Simulation Initialization. Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC). Orlando, FL.

Carr, F., Henninger, A., Hallenbeck, P., and Morse, J. (2007). Joint Data Alternatives Study. MITRE Technical Report.

Collin, S. (2002). Dictionary of Computing, 4th edition, Peter Collin Publishing Ltd.

Connors, P., Trnka, G., Hallenbeck, P., Allitt, E., and Hartel, R. (2006). Outbrief from the Army Initialization Workshop, October, 2006. Ft. Monmouth, NJ.

Cooper, D. and Clancy, M. (1982). Oh! Pascal! W.W. Norton & Company, Inc. New York, N.Y.

Frambes, T. (2005). Army Battle Command System Functions, Integration, and Parallel Support of the Military Decision-Making Prociess. Master's Thesis U.S. Army Command and General Staff College.

Greene, H. and Mendoza, R. (2005). Lessons Learned From Developing the ABCS 6.4 Solution. Defense Acquisition University Journal. Alexandria, VA.

Moore, D. (2007). Battle Command Software: Meeting the Commander's Needs? USAWC Strategy Research Project and Master's Thesis. U.S. Army War College. Carlisle Barracks, PA.

Prochnow, D., Fogus, M., Vintilescu, J., Borum, B. (2005). Initialization of Distributed Simulations: A Better Way? 2005 Fall Simulation Interoperability Workshop. Orlando, FL.

Sprinkle, R. and Black, C. (2006). Joint BC & Simulation Systems Initialization Process. Spring Simulation Interoperability Workshop. Orlando, FL.

Sprinkle, R., and Brown, R. (2004). Capability-based Development of Future C4ISR Systems. Spring Simulation Interoperability Workshop. Arlington, VA.