1999 Spring Simulation Interoperability Workshop
Orlando, Florida

Data Alignment between Army C4I Databases and Army Simulations

Michael R. Hieb, Ph.D.
AB Technologies, Inc.
1901 N. Beauregard St.
Alexandria, VA 22311-1705
(703) 575-2800
mhieb@abtechnologies.com

Major James Blalock
Data Management Branch
ODISC4
400 Army
The Pentagon
Washington, D.C. 20310-0400
(703) 697-3131
blalockj@hqda.army.mil

Keywords:
Army Battle Command Systems (ABCS), Joint Common Data Base (JCDB), JCDB Transformational Data Model (JTDM), Command, Control, Communications, Computers, and Intelligence (C4I) Interface, Defense Information Infrastructure Common Operating Environment (DII COE), High Level Architecture (HLA), Joint Technical Architecture-Army (JTA-Army)

ABSTRACT: *C4I Interfaces to Simulations are limited in functionality. One of the principle factors causing this is a lack of interface standards. While standards are a necessary condition, they are not sufficient. In order to have a complete interface, the data to be exchanged must be represented in both systems (C4I and Simulation). However, the current status is that C4I systems and simulations 1) do not contain the same data elements; 2) represent data differently, and 3) are often unaware of the other's data requirements. The Joint Technical Architecture-Army (JTA-Army) mandates use of specific data models for certain classes of information systems (such as tactical C4I systems). Simulation developers should be cognizant of these data models in the development of new Army Simulations.*

The Army is using a common data base, the Joint Common Data Base (JCDB,) in it's Army Battle Command Systems (ABCS). In order to interface effectively to Army C4I systems, simulations must represent the critical data elements that are 1) used to initialize the JCDB; and 2) sent between the simulations and C4I systems. However, often Army simulations do not contain these critical data elements which will result in functionality that must be provided by interfaces. Each data element that does not have an exact match on the simulation side causes a translation/transformation to occur, with resulting cost and complexity. Since any interface must align any differences, the interface can become quite complex.

This paper compares an Army C4I Data Model of the JCDB to a simulation representation (an Army Modeling and Simulation Office proposed standard that is representative of current and future simulations) to identify areas which are not aligned. Results of this analysis show that current Army representations are not aware of the standard data models that will be used in future Army ABCS systems.

1. Introduction

To date, the development of C4I to Simulation interfaces has been problematic. Most existing C4I interfaces to Simulation have been developed as a separate component added on after initial Simulation development and typically handle a small subset of the messages or data necessary for interoperability. Significant human intervention is needed to achieve realism for the training audience in an exercise. Simulation systems rarely handle free text messages or consider how a message is carried (communication effects). C4I systems are subject to different design constraints than Simulation systems, in areas such as reliability, communications and operator interaction. All of the above factors have combined to produce C4I interfaces with more or less limited functionality.

There are three areas that must be addressed to achieve "seamless" Simulation to C4I interfaces. The first area is interface standards for the entire range of information exchange. The development of these standards is addressed by Hieb & Staver [4], supported by an Interface Technical Reference Model [2], [4]. AMSO is playing an active part in establishing these and other simulation standards for the Army [1], [8]. The second area is common software components. The Defense Information Infrastructure Common Operating Environment (DII COE) provides an example of this. Timian, et. al. [10] discusses how the Army will use certain DII COE components (such as the DII COE Communications Server and the COE Message Processor) in future C4I interfaces. The third area is alignment of data models, which is the subject of this paper.

All future Army Information Systems must be developed under the Army Enterprise Architecture, as described in the Operational, Systems and Technical Architectures. The Joint Technical Architecture (JTA) is the Department of Defense (DOD) set of standards that must be adhered to. The JTA-Army [6] sets forth the standards for the Army, and future tactical systems (reference Section 4.2.2 -- Data Model) must be based upon the C2 Core Data Model (C2CDM). All of the ABCS systems will use the C2CDM-compliant Joint Common Data Base (JCDB).

Simulations have traditionally used highly specialized knowledge representations to achieve acceptable runtime performance. Standard representations have been slow to emerge due to the differing system components of simulations (software and hardware). The High Level Architecture addresses interoperability between simulations for certain classes of information through specification of a Federation Object Model (FOM) for federations of simulations, and is mandated for all future DOD Simulations.

Thus, there are standard architectures that have been created, to engineer interoperability in the C4I and Simulation domains. There is, however, no standard architecture for interfacing simulations to C4I equipment. Current C4I interfaces for Simulation are hindered due to this lack of architecture standards. Hieb and Staver [4] and Fournoy [3] discuss interoperability solutions based on interfacing the standard architectures (e.g. HLA & DII COE).

Simulations use many different representations that are analogous to data models. In the case of common Army simulations, these representations in many instances, do not align with, or map to, the JCDB. If there is a mismatch between the Simulation and C4I standards, software translators will have to be built to align the data. Such translation software and associated interfaces are very costly and reduces interoperability through lack of functionality. In addition, if data elements are missing in simulations that are utilized in real world systems, interfaces become much more complex, as they must both create and synchronize such data.

In this paper we evaluate a small portion of the JCDB to a simulation representation standard for a *Unit*. Since AMSO is proposing various Object Model Standards for Army Simulations, we use their Unit Object Model. It was designed to accommodate many of the current and future Object Oriented simulation representations and is representative of current Army simulations.

A note about terminology. There are different interpretations of "data" for C4I systems and simulations. C4I systems typically use highly structured "Data Models" that are expressed in a formal language (e.g. IDEF1X). The "database" is a physical implementation of the Data Model. In this paper we use the term "JCDB" to refer primarily to the Data model (which is the *JCDB Transformational Data Model*). Simulations typically have not expressed their "Data Models" in formalized notation. Instead simulations use various representations to support their functionality. Data may be expressed in data structures, rules or objects. All future simulations are using "Object Models. Data models and simulation representations are further discussed in sections 3 & 4, respectively.

The remainder of this paper is organized as follows. Section 2 describes the different approaches to data interoperability that the simulation and C4I worlds are taking and how this affects C4I interfaces. Section 3 describes an Army Data model used for its ABCS systems. Section 4 describes an AMSO Unit Object Model. Section 5 compares the two representations and Section 6 draws some conclusions for C4I Interfaces.

2. Different Approaches to Interoperability

Why do our current C4I Interfaces fail? The basis of the problem is that we do not represent the same information (data) in both systems. If we do not have information about an entity in a simulation that a C4I system needs, no interface can ever be created to transfer or translate

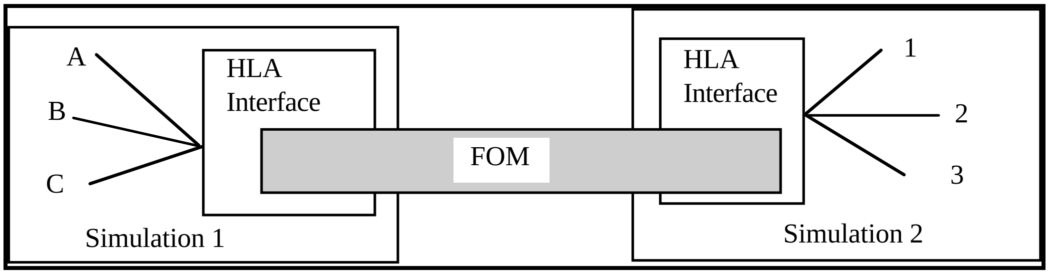

Figure 1: Simulation Interoperability though Information Transfer

this data. This data must be created by an interface. If data is seriously misaligned, then the interface must be made much more complex to perform the alignment. However, if the simulation is designed with the C4I system's data model in mind, the interface is much simpler, and can be made more like the actual information transfer between C4I systems.

Previous interface experiments, such as the Modular Reconfigurable C4I Interface (MRCI) program by the Defense Modeling and Simulation Organization showed that it is possible to translate message formats into the FOM format [5], [7]. However, this translation is complex due to the ambiguity of message formats. One of the main lessons learned from the MRCI program was the desirability of using more structured and flexible representations.

As an example, take the notion of a person's Social Security Number. This is a common identifier used in ABCS systems (e.g. logistics) for tracking people, and is an attribute of the JCDB Person Entity. However, few simulations, even entity level simulations that represent dismounted soldiers, use a Social Security Number to designate a "person" entity. Instead these simulations typically use a form of ID unique to their simulation. If these simulations are used to stimulate Army C4I systems, used to track and manage people, then there will be several transformations that simulation IDs must go through to be put into "real" data that is valid for the JCDB Person Entity. While this may seem to be a trivial problem, as we have seen with the Y2K problem, limiting representations can have a profound impact later. Mapping from a alphanumeric string or a 5 digit integer (or however the simulation represents entity IDs) into the 9 digit integer format of a Social Security Number, and keeping these in synchronization requires custom software and limits functionality. Social Security Numbers could be used in a simulation easily in order to: 1) transfer Social Security number data from the JCDB to a simulation during scenario generation, so that real data can be used for execution; and/or 2) to create valid Social Security numbers for simulated entities, in order to populate the relevant portions of the JCDB with exercise specific data.

The primary argument of this paper is that we need to have simulation representations that are better aligned with the emerging C4I data models. In the past we have concentrated on building software interfaces that translate formats, but have deferred the more fundamental problem of moving towards the same representation in both simulation and C4I systems. This issue is illustrated in Figures 1-4.

There are two different conception of interoperability for C4I and Simulation Systems respectively. Figure 1 shows simulation interoperability through use of the HLA. The HLA uses a shared data model for data exchange - the FOM. Each simulation utilizes it's own internal representation. Thus data element "A" in Simulation 1 may map to FOM attribute/parameter "Y", and data element "1" in Simulation 2 may map to the same FOM element "Y". So that "A" and "1" represent the same "data", but are different in syntax.

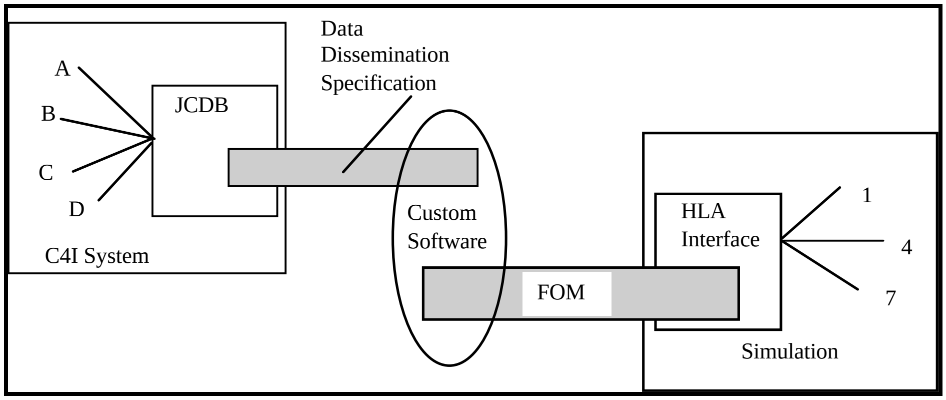

Figure 3: Simulation Interface to C4I though Software

Figure 2 shows the vision of ABCS system interoperability, with all C4I systems sharing a common data base, and exchanging data to keep their data bases synchronized. Each system in addition will have it's own unique data to satisfy it's unique functions, which is in it's data base but not shared, as the common data is. In this stricter implementation of interoperability, the systems all use the same data model internally, rather than using a shared data model for data exchange. If C4I system 1 modifies data element "A", this is reflected in C4I system 2's data base through data dissemination. No transformations are necessary.

Figure 3 shows the one possibility of using the JCDB in interfaces. The exact interface mechanism is unimportant, as the point is that the data model is not aligned with the simulation representation. So C4I system data element "D" has no representation in the Simulation, and data element "C" must under go transformation to be expressed as attribute "7" in the simulation. The interface will be driven by the mismatch between the two data models.

Figure 4 shows how the interface could look if the data models were aligned. They need not use the same syntax, but would express the same data elements, with a compatible organization of data. Thus "A" is the same data element as "*A*", with the same name, meaning and units/enumerations, but expressed in a different representation (e.g. objects rather than relational data tables).

3. C4I Data Models

Today, battlefield information exchange is accomplished primarily by sending messages. The definition and documentation of these messages are provided by various messaging standards, such as Variable Message Format (VMF), and the U.S. Message Text Format (USMTF). Each message standard provides a means to define message form and functions (i.e., transfer syntax), that includes the definition of the message fields that are contained in each message. The message fields, which are currently defined in the various message standards,

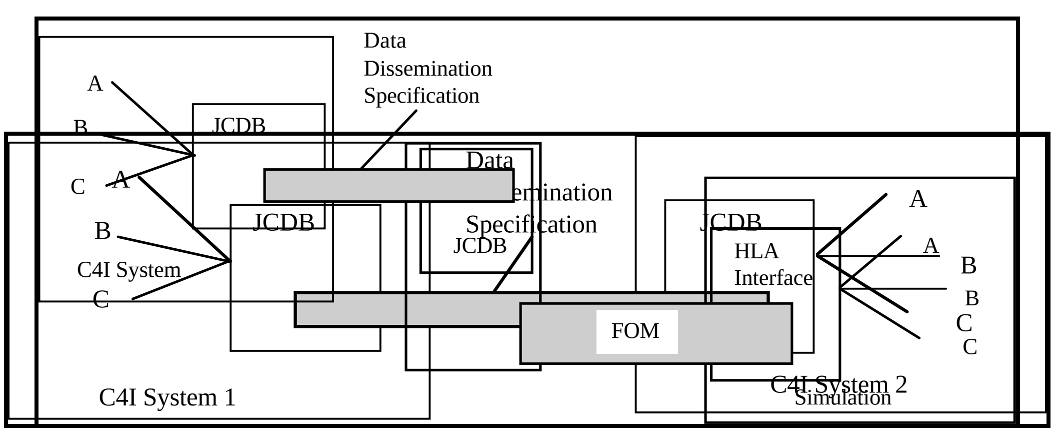

Figure 2: C4I Interoperability through Common Data Models / Figure 4: Simulation Interface to C4I through Data Models

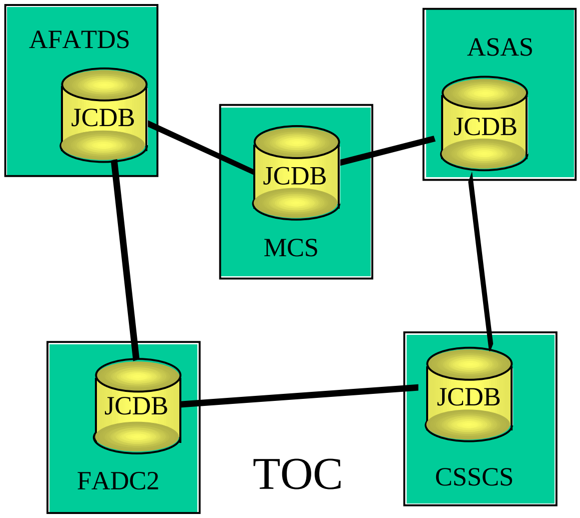

Figure 5: Use of Common Data Models for ABCS

are not mutually consistent across message types, nor are they based on any process or data models, either within a message system or across message systems.

Newer techniques can provide direct database-to-database exchange of data, without the user having to follow a rigid format. To use these newer techniques, the message fields must be converged with the data element set that is developed through activity and data modeling efforts defined in the JTA-Army. This set is compliant with the DOD data element standards established in accordance with the DOD 8320.1 series of directives.

3.1 Data Modeling

Developing an effective information exchange interface with/to Army C4I databases requires and understanding and utilization of the database structures. Following common data modeling practices is essential for the proper identification and structuring of the information to be exchanged between the C4I and simulation domains. To support the identification of information and information interchange requirements, the DOD has selected the Integrated Computer Aided Manufacturing Definition (IDEF) modeling method. Activity modeling is covered under IDEF0 standards, while data modeling is covered under IDEF1X standards. Use of the IDEF standards allows definition of:

- Symbols (i.e., syntax) associated with modeling concepts and ideas.
- Rules for composing these symbols into abstract constructs.
- Rules for mapping “meanings” (i.e., semantics) to these constructs.
- Definitions of the relationships between activities and entities.

In order to provide a single authoritative source for data definitions, the DOD created the Defense Data Dictionary System. The DDDS, is a DOD-wide central database that includes standard data entities, data elements and, soon, data models. The DDDS is used to collect and integrate individual data models into a DOD enterprise data model and to document content and format for data elements.

Recent studies show three necessary data characteristics must be known to define interoperable databases. These characteristics are contained within the combination of the Defense Data Dictionary System (DDDS) and IDEF0 & IDEF1X models.

3.2. Army C4I Data Models

The JTA-Army addresses both message-based and direct database-to-database exchange of data. However, future tactical systems within the scope of the JTA-Army (reference Section 4.2.2 -- Data Model) must be based upon the C2 Core Data Model (C2CDM). In addition, the Variable Message Format (VMF); US Message Text Format (USMTF), and the Tactical Digital Information Link (J Series) Message Formats are mandated in Section 4.2.4 -- Data Exchange, but *only until mechanisms that use standard data elements are approved.*

Using a common data model such as the JCDB has many benefits, including:

- Enhanced Interoperability
- Efficient DB to DB exchange of data
- Effective use of Standard Data Elements
- Increased Data Integrity
- Increased flexibility
- Reduced burden to operators
- Reduced maintenance costs
- Reduced use of formatted messages

Future ABCS systems will all use a common data model for any shared data, the JTDM. Figure 5 shows a conceptual view of how these systems will interoperate. Ability to use the JCDB will allow C4I interfaces to access all of the ABCS systems in a standard way.

3.3 JCDB Description

There are 293 tables in the current version of the JCDB, with an associated set of 1244 owned attributes (fields).

The Army Common Data Base (ACDB) was renamed the Joint Common Database (JCDB) in its last release. This decision was stimulated by on-going work with DISA, and the incorporation of enemy tables and data elements from the MIDB (a DIA program) into the ACDB.

The current JCDB version (3.3b) provides the database schema and implementation information to support the ABCS 5.0 Synchronization Event. 55 tables will be used for the minimum functionality requirements for ABCS 5.0 as well as 89 look-up or reference set tables that support the 55 primary tables.

The JCDB is comprised of several different components:

JCDB Transformation Data Model (JTDM) – The JTDM is an IDEF1X (logical) Data Model of all Army and Joint shared data from which the Joint Common Database is derived. The JTDM was developed by the PEO C3S Horizontal Technology Integration Office and is based on the C2 Core Data Model as mandated by the JTA and JTA-A.

Joint Data Dictionary (JDD) – The ADD is a dictionary of all data elements set fort in the ABCS Common Database. It includes the data element name, definition, datatype and domain values/enumerated types for each data element. Use of the ADD results in the use of a common language by all ABCS systems.

Joint Common Database (JCDB) – The ACDB is the (physical) database of all ABCS shared data and is derived from the JTDM. The JCDB uses the standard elements set forth in the ADD. The JCDB is currently represented in both Informix and Oracle RDBMS formats.

The JCDB provides a C2 Core compliant database of *Shared* Battlefield Data which:

- supports capture of friendly and enemy activities, strength, status, estimated/current capability and location;
- provides for capture of consumption and environ-mental factors to support Course of Action analysis;
- supports capture of materiel status and location;
- supports target nomination;
- supports evaluation and verification of reported information;
- supports development of Operational Orders and Plans.

ORGANIZATION

ORGANIZATION identifier
ORGANIZATION principle equipment type
ORGANIZATION reference number
COUNTRY code (FK)
ORGANIZATION unit identification code (IE1.1)
ORGANIZATION parent unit identification code
ORGANIZATION functional area identifier
ORGANIZATION name
ORGANIZATION formal abbreviated name
ORGANIZATION-TYPE identifier (FK)

Figure 6: JCDB Organization Table

ORGANIZATION-TYPE

ORGANIZATION-TYPE identifier
ORGANIZATION-TYPE category
codeORGANIZATION-TYPE echelon code
ORGANIZATION-TYPE arm code
ORGANIZATION-TYPE function code
COUNTRY code (FK)
ORGANIZATION-TYPE sevice code
ORGANIZATION-TYPE mobility code
ORGANIZATION-TYPE operational mode code
SYMBOL basic display code (FK)

Figure 7: JCDB Organization Table

The concept of "perception" has been adopted from the International ATCCIS Generic Hub Data Model. This table captures metadata about information from other dynamic tables in the JCDB.

3.4. ACDB Organization Entity

There are 5 basic types of entities in the JCDB: Organization, Feature, Material, Facility and Person. The concept of an organization is needed to structure information. The definition of an organization is an administrative and/or functional structure that has personnel and equipment assigned to it to accomplish a specific mission. The Organization Identifier provides the basis for Task Organization, Common Operational Picture, and Situational Awareness.

The JCDB is very large and it is only possible to show in detail a few of the Organization Tables. Figures 7 shows the Organization table, with Organization Identifier as the primary key and numerous other attributes, with Country Code (giving the unit affiliation) and Organization Type ID as foreign keys. The JCDB also shows numerous relationships to other tables such as Organization Type and Organization Capability. Figure 8 shows the Organization-Type table, with its attributes.

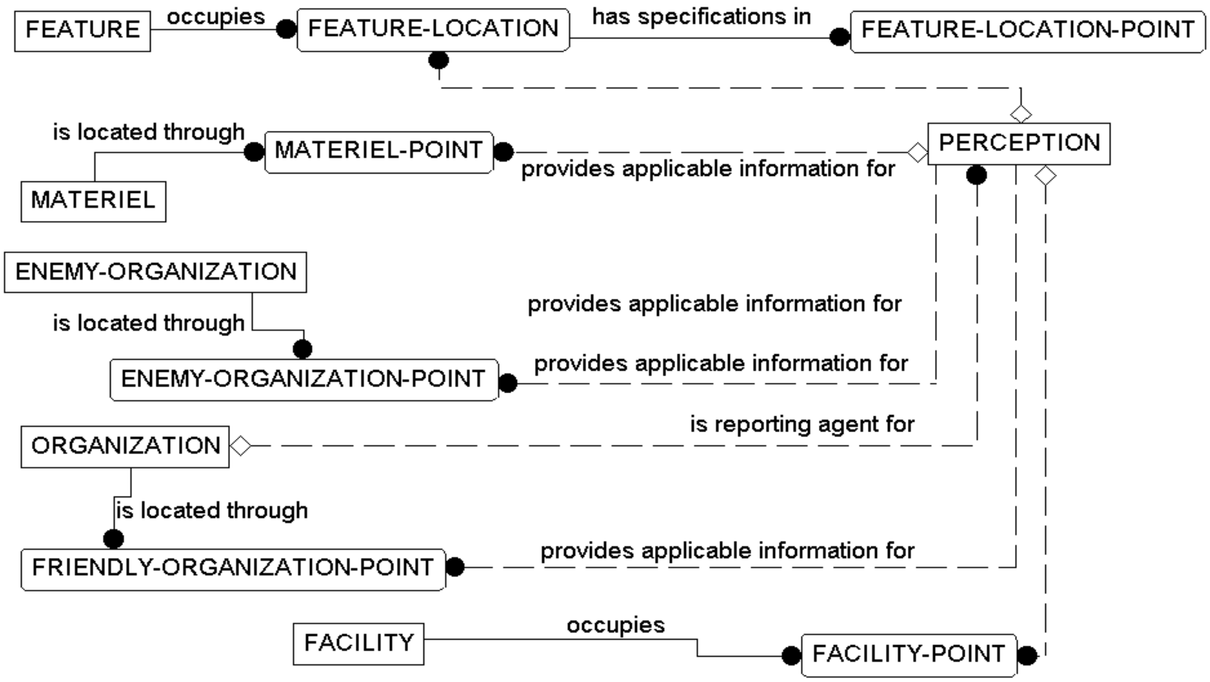

Figure 8: JTDM View of Unit Status

As an example of how the Organization Construct is used, Figure 9 shows a Battlefield Object-Point View. This view captures the specification of position/location of battlefield objects. This View consists of a various types of points, their related battlefield objects, and PERCEPTION. An Organization entity is associated with a Friendly-organization Point, which gives the location, as will be seen in more detail in Section 5.

4. Army Simulation Representations

Object-oriented programming offers the potential for increased code reuse, maintainability, and ease of developing simulations. Because of these perceived benefits, simulations are increasingly using object-oriented. Relational models can be transformed into object oriented models.

In order to prevent duplication of effort and the development of incompatible models the Army is developing an Army object management initiative. This initiative will document the standard Objects that define the minimum set of objects and object methods needed for the development of Objects in models and simulation.

4.1 AMSO Object Oriented Simulation Standards

Many of the current Army and Joint model development efforts have embraced the use of Object Oriented Programming for their model development efforts. As a result, there has been a proliferation of competing object models. The Army Object Standards focus on a high-level object class structure, independent of any specific simulation environment. This allows Simulation developers to tailor the high-level object standards to their specific applications through lower-level class/ instantiations that extend the standards to a specific Simulation requirement. The overall impact in the development of standard abstract objects will be to organize future Simulation along a common object structure to support interoperability, object reuse, and community understanding of the Simulation. AMSO formed the Object Management Standards Category (OMSC) in April 1997 to initiate the proposed policy.

Several Object standards have been proposed including ones for Unit, Platform and Logistics.

4.2) Unit Object Oriented Standard

The Unit Standard is shown in Figure 8 and described in an AMSO Army Standard Unit Object technical report [x]. The standard relies upon methods to encapsulate specific data formats. Thus, instead of specifying a coordinate system for location, there is a function call to “getLocation”. The Unit Class is the base class, with several associated subclasses.

5) Alignment Experiment

In order to assess the current situation regarding alignment of data models, we matched the AMSO Unit Standard to the JCDB. We choose to use the Unit Standard as the base since it is much smaller.

The experiment matched the base class for Unit to the Organization table in the JCDB. This is shown in Figure 9. The methodology used was as follows. If there was a match for a unit class attribute in the organization table, a match type of 1 was given. If there was no match of type 1, then we looked in the rest of the JTDM for the attribute. If we found it, we gave it a 2 if it was an exact match. If no identical names were found, we matched the definitions, and gave a match type of 3. If the definition was close, but not exact, but a match could be obtained through a transformation, we gave a type of 4. And if there was no match for a JCDB organization attribute, we gave a match type of 5.

The results were that out of the attributes in both representations, there were no direct matches on name. There were several definition matches (type 3): for speed and movement direction (but they were not in the Organization Table, but instead in the Friendly-Organization-Point table which has Organization: identifier as a foreign key); Country:code matches to getSide; and Organization-Type:echelon matches to getEchelon. Most of the Unit Class attributes were only able to be matched through utilization of a transformation. For example, the getLocation attribute matches to location attributes in the Friendly Point Table, but there are two coordinates in latitude and longitude. The other matches of type 4 were given due to the use of enumerations in the JCDB which are assumed to be different than what would be used in the Unit Class and are not specified. Most of the attributes in the Organization Table did not have a match. The majority of the attributes did not have a direct match.

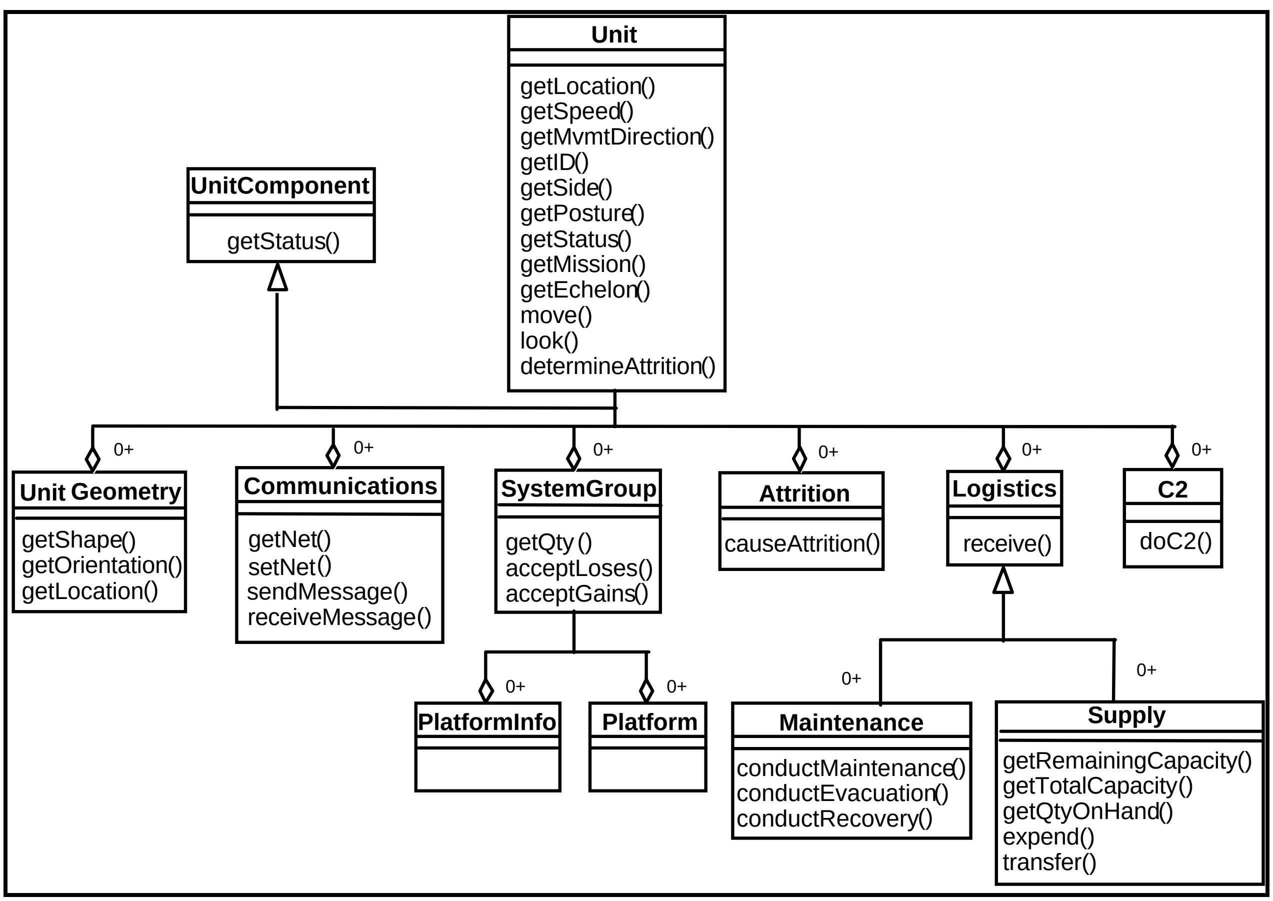

Figure 9: Proposed AMSO Unit Object Standard

AMSO Standard	JTDM	Match
Unit: getLocation	Friendly-Point:lat coor/long coor	4
Unit: getSpeed	Friendly-Organization-Point:speed rate	3
Unit: getMvmtDirection	Friendly-Organization-Point:bearing angle	3
Unit: getID	Organization:identifer	4
Unit: getSide	Country:code (FK)	3
Unit: getPosture	Organization-Type:operational mode code	4
Unit: getStatus	various	4
Unit: getMission	Organization-Mission: Mmssion code	4
Unit: getEchelon	Organization-Type: echelon code	3
Unit: Move		n/a
Unit: look		n/a
Unit: determineAttrition		n/a
	Organization:principle equipment type	5
	Organization:reference number	5
	Organization:unit identification code (IE 1.1)	5
	Organization:parent unit identification code	5
	Organization:functional area identifier	5
	Organization: name	5
	Organization:formal abbreviated name	5

Figure 9: Match of AMSO Unit Object Standard to the JCDB Organization Entity

Space does not permit presentation of the full analysis, but the result is similar as above for the remainder of the unit subclasses, with approximately 25% indirect match (type 3). All of the status methods from the Unit Class have matching data elements in the JCDB, the converse is not true. There are many more detailed data elements in the JCDB for Unit (Organization) related tables. This experiment was also performed on the Platform Object standard with similar results.

The implications are that the simulation representation will not support initializing a C4I system, since it does not have the representation to do so. If a simulation using this representation is initialized from a C4I system using the JCDB, then custom interface software must be written to translate from the JCDB data, formats, and names to the simulation representation.

6. C4I Data Models in Simulation Standards and Architectures

Our investigation into the alignment between a standard Simulation Object Model (representative of current simulation representations) and JCDB Data Model that will be used in future Army Tactical C4I systems shows discrepancies in several areas. If these discrepancies are not addressed by future Army simulations, required simulation interfaces to ABCS systems will be costly and, in certain cases, ineffective.

In the past simulations have not been able to obtain explicit representations from real world systems. This is now changing, and should enable interfaces to become "thinner" and more transparent. Certain classes of simulations will not be affected by these C4I data models, but most will find it necessary to take them into account to model communications, information warfare and other effects caused by the use of C4I systems.

7. Acknowledgments

The authors thank LTC Joel Cromwell for his assistance on the Data Modeling aspects of this paper, the many other people that have improved this paper through their critical reviews. Dr. Hieb was supported by ODISC4 of the Army while writing this paper.

8. References

[1] Army Modeling and Simulation Office: “http://www.amso.army.mil”, 1998.

[3] Carr, F.H. and Hieb, M.R.: “Issues and Requirements for Future C4ISR and M&S Interoperability”, 7th Conference on Computer Generated Forces and Behavioral Representation, 1998.

[3] Flournoy, D: “Reconciling Emerging Infrastructure Standards to Promote C2-to-Simulation Interoperability”, Paper 98F-SIW-027, 1998 Fall Simulation Interoperability Workshop, 1998.

[4] Hieb, M.R. and Staver, M.J.: “The Army’s Approach to Modeling and Simulation Standards For C4I Interfaces”, Paper 98F-SIW-259, 1998 Fall Simulation Interoperability Workshop, 1998.

[5] Hieb, M.R.; Cosby, M.; Griggs, L.; McKenzie, F.; Tiernan, T.; & Zeswitz, S.: “MRCI: Transcending Barriers between Live Systems and Simulations”, Paper 97S-SIW-197, 1997 Spring Simulation Interoperability Workshop, 1997.

[6] Hodge, D. and Bradley, B: “Army Standard Unit Object”, Technical Report TR-634, US Army Material system Analysis Activity, July, 1998.

[7] Joint Technical Architecture – Army, Version 5.0, Office of the Director, Information Systems for Command Control , Communications and Computers, http:// www.hqda.army.mil/techarch/, September, 1997.

[8] Lightner, M., Schanduaa, J., Cutts, D., & Zeswitz, S.: “The High Level Architecture Command and Control Experiment – Lessons Learned in Designing an Extended Federation”, Paper 98S-SIW-93, 1998 Spring Simulation Interoperability Workshop, 1998.

[9] McGlynn, L. E. & Timian, D.: “Army Model and Simulation Standards – Tools in the SBA Kit Bag” Paper 98F-SIW-265, 1998 Fall Simulation Interoperability Workshop, 1998.

[10] McKenzie, F & Risner S.: “Joint Simulation System (JSIMS) Approach to C4I System Interoperability” Paper 98F-SIW-117, 1998 Fall Simulation Interoperability Workshop, 1998.

[11] Timian, D., Hieb, M.R., Glass, J. and Staver, M.J.: “Using Standard C4I Components to Interface to Simulations”, Paper 98F-SIW-035, 1999 Spring Simulation Interoperability Workshop, 1999.

Author Biographies

MICHAEL HIEB is a senior scientist at AB Technologies. Dr. Hieb was the architect of the Modular Reconfigurable C4I Interface (MRCI) interface project while at SAIC. He received his PhD in Information Technology at George Mason University in 1996, developing an instructable ModSAF agent. He performed his doctoral research at the Center for Excellence in Command, Control, Communications and Intelligence at GMU. He has published over 30 papers in the areas of learning agents, knowledge acquisition, interface technology and multistrategy learning. When working for IntelliTek, Dr. Hieb implemented a distributed problem-solving testbed at the Goddard Space Flight Center. Previously, he worked as a Nuclear Engineer for General Electric.

JAMES BLALOCK is a Major with the U.S. Army and a staff officer in the Architecture Directorate, Standards and Strategy Division of the United States Army Office of the Director of Information Systems for Command, Control, Communications and Computers (ODISC4). He coordinates Data Architecture standards for the Army. In his capacity as the Modeling and Simulations officer for ODISC4, he oversees the Modeling and Simulation efforts for the Army Enterprise Architecture. His previous assignments included serving as a Systems Acquisition Officer and Technical advisor with the CECOM Systems Management Center, for the SOUTHCOM Relocation Project; and a staff officer with the CECOM RDEC Special Projects Office.