9th International Command and Control Research and Technology Symposium
Coalition Transformation: An Evolution of People, Processes and
Technology to Enhance Interoperability

Managing Data for Interoperability:
The Army C4ISR and Simulation Initialization System (ACSIS)

Randy Shane, Knowlegeable and Innovative Technical Solutions LLC
Phil Hallenbeck, MITRE Corporation

Whitfill Central Technical Support Facility
North Avenue and 53rd Street
Fort Hood, TX 76544

254-532-8321 x2664

Randy.Shane@hood-ctsfmail.army.mil
Phil.Hallenbeck@hood-ctsfmail.army.mil

Abstract

Network-centric warfare operations rely on a large amount of data shared among most or all nodes in the network—a phenomenon often referred to as the "common picture." Critical to achieving this common picture is the initialization of these nodes with very large amounts of relevant, correct, and consistent data. Existing initialization processes, however, have been built for individual systems, neglecting the need for interoperability among a network of systems on the battlefield. The result is often the inability of systems to communicate with one another, or—just as perilous—operator confusion when one system cannot reliably interpret the data sent by another. Existing initialization systems also lack scalability, breaking down when tasked to rapidly produce large product sets for major operations.

The Army Command, Control, Communications, Computers, and Intelligence Surveillance and Reconnaissance (C^4ISR) and Simulation Initialization System (ACSIS) provides timely, consistent, and verified initialization data to C^4ISR and simulations systems. ACSIS does this by providing a rigorously controlled, centralized repository for shared data and a common set of tools and interfaces from which systems can initialize. ACSIS provides a data model that can be easily updated to allow for new target systems, and which is robust and complete enough that it could be used as a multi-service repository of initialization data, or perhaps for multinational operations.

ACSIS has been and is being used to develop initialization dataloads for Army systems in Operations Enduring Freedom (OEF) and Iraqi Freedom (OIF).

Introduction

The problem of initializing automated systems has probably existed as long as have the systems themselves. Histories of the likes of ENIAC and other early computers describe the efforts undertaken to program them not only with the computing logic employed to solve mathematical problems, but with the "input data" describing the problems themselves.[1] In like manner, the problem domain of military command and control has probably always dealt with the question of common information[2] from which staff officers and commanders can work and make decisions. The rapidly expanding field of network-centric warfare (NCW), however, greatly exacerbates these historical problems—from that of initializing single systems to initializing thousands, and from information commonly understood by trained staff officer to information understandable by software of widely varying quality and sophistication. Perhaps nowhere are these problems seen more than in the United States Department of Defense as it undergoes its widely-publicized "transformation" to a "21st-century force" enabled by networked information systems.

This paper describes a working prototype system, the Army Command Control Communications and Computers Intelligence Surveillance and Reconnaissance (C4ISR) and Simulation Initialization System or ACSIS, which can help solve this problem for

DoD and potentially for multi-national forces as they struggle to harness the power of the network. The reader will see that while ACSIS is by no means a complete solution to the initialization problem, it represents an important step forward, and lessons learned by the ACSIS team may be important for other teams as they design and implement automation systems for their respective defense organizations. A schematic of ACSIS is shown at Figure 1.

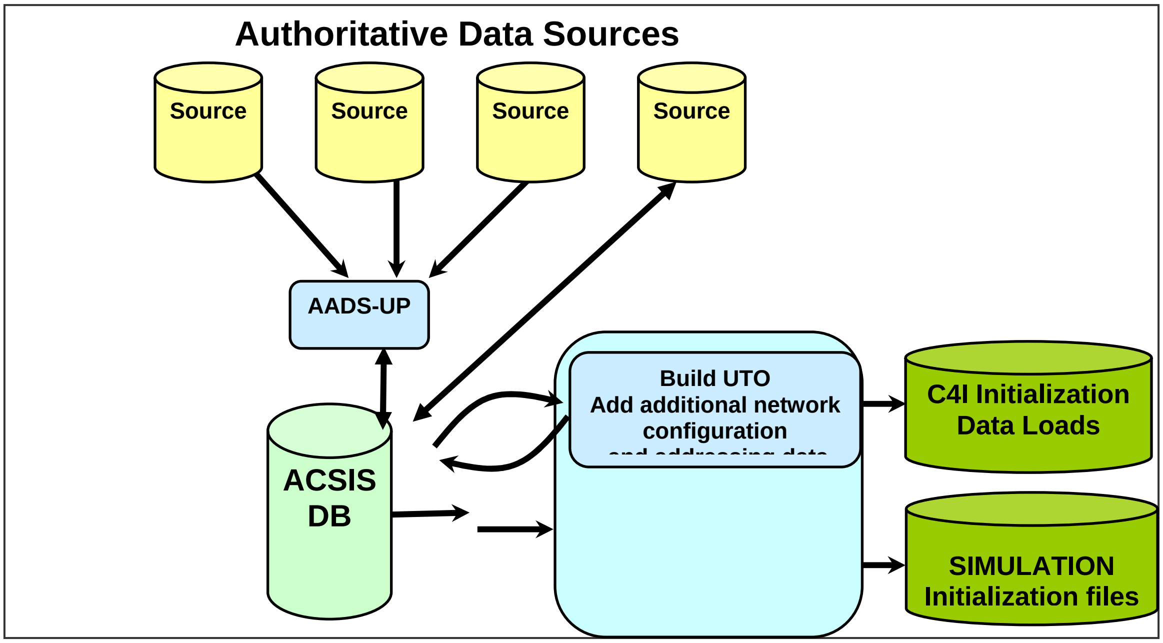

Figure 1. Army C4ISR and Simulation Initialization System (ACSIS) Schematic.

Broadly speaking, the initialization problem domain can be broken down into four major areas:

- Standardization of data kept in service or national repositories
- Standardization of data used by automation systems, and the creation of a repository or repository system to house this data
- The creation of tools and processes to initialize systems using data from the repository, and
- Transition of the repository and tools to versions which are easily accessible and usable by warfighters (not engineers at some distant site) to provide a truly responsive capability to update initialization data as required.

We will address each area in turn, but first will present a brief history of ACSIS and the US Army initialization problem as seen by ACSIS' creators.

Early US Army Initialization Efforts and the ACSIS

As the US Army began its efforts towards "digitizing" its units in the middle and late 1990s, system initialization efforts (not surprisingly) focused on the same bottom-up methodology as did the experimentation efforts themselves. Efforts to introduce automation systems into combat and tactical vehicles in platoons and companies led to the creation of data products focused on these small units. The process for building these products centered on the database for Force XXI Battle Command Brigade and Below (FBCB2). This product, which was built by a contractor team from TRW (now Northrup Grumman Mission Systems or NGMS), also served to initialize many routers and communications systems in the then-fledgling Tactical Internet (TI). Using the FBCB2 database as a source, initialization products were then built for the other systems in the TI, such as the Maneuver Control System (MCS), the Advanced Field Artillery Tactical Data System (AFATDS), and the All-Source Analysis System (ASAS, an intelligence analysis system).

At the same time, the Army programmed its efforts towards the acquisition of a "system of systems," the Army Battle Command System or ABCS, representing a major change from the then-existing approach of disparate and disconnected individual systems. Few knew at the time, however, how difficult it would be to transition from individual systems to a system of systems, particularly in the area of initialization.

Early in the development of ABCS, the Army created the Central Technical Support Facility (CTSF) at Fort Hood, Texas, and charged the new organization with the software and systems integration of ABCS. An integral part of this role was the creation of initialization products for the systems to be integrated. As the engineering team at the CTSF worked to create a viable initialization system for ABCS, numerous problems became apparent.

First was the difficulty and uncertainty in creating a systems and network architecture that would support the hundreds of information exchanges envisioned for the TI. Associated with this problem were the problems of allocating network identifiers, such as IP addresses and system names, to support efficient information flow, and of creating an understandable view of the architecture from which engineers could build data products.

Next came the sheer difficulty of creating a synchronized product set for hundreds or even thousands of systems, each of which required initialization with dozens or even hundreds of data items. Even finding the source data itself for these data items was a challenge, made more severe by the lack of documentation of the processes and sources for initializing each system.

Third was the disparate nature of the data products for different systems: Not only data models but the data items themselves differed widely between systems, with different systems using different data items to describe the same thing. For instance, several name formats existed in the early TI depending on the system in use.

All these difficulties resulted in yet another problem—that of quickly creating data product sets for a given exercise or operation. The initialization process in use, initially used for small experiments, did not scale well at all, and product sets literally took months to create for brigades or larger units. This portended severe problems in quickly deploying a “digitized” force at the very time the Army was trying to dramatically improve its deployability.

In response to its perception of these problems and to fulfill its responsibility for initializing ABCS, in early 2002 the Systems Engineering (SE) team at the CTSF created what was then called the Repository of ABCS Data or ROAD. ROAD was intended to provide a single repository of data from which ABCS systems could be initialized using a standard tool set that created a “tailored” database load for each destination system.

While the Army’s digitization efforts were quite successful by world standards, culminating in the 2001 Division Capstone Exercise (DCX), further problems loomed in the area of initialization. Multi-service exercises in 2002, such as the Millennium Challenge exercise in the summer of that year, showed that ABCS had difficulty communicating with systems from other services. One of the key problems was initialization. Further, developments within the Army itself, as units procured their own automation systems in advance of the fielding of ABCS, complicated the issue by introducing purely-commercial products and standards in areas such as electronic mail (e-mail), web services, and directory services—areas in which ABCS, and the initialization of it, was very different from the commercial world.

As these lessons were learned and the United States’ commitment to the global war on terrorism mounted, it appeared a much more comprehensive initialization system than ROAD would be required. Hence, the CTSF SE team began work to expand the data model and tool set from ROAD, to coordinate more fully with authoritative data sources, and to gain knowledge on initializing the simulations on which much of training and mission preparation are now based. This project became known as ACSIS, and the overall effort in the Army and DoD gained the title of Initialization Capability or IC.

As ACSIS gained momentum and support from the Army digitization community at the CTSF, the team was very fortunate to have the active support of the Army’s and DoD’s simulation communities, which not only stood to benefit greatly from the project (since the initialization of simulations also presented acute problems to DoD) but had also gained significant experience in this area from their own earlier efforts. Indeed, the ACSIS team often found that problems it encountered such as in data modeling had already been seen and addressed by simulations engineers. This resulted in a strong partnership, which endures to this day and has every prospect of continuing.

The IC Problem and the ACSIS Response

The problem encountered and addressed by the ACSIS team consists of a fragmented, “stovepipe” initialization system originally built to initialize individual systems with no thought to a system of systems—a condition that in many if not most parts of DoD persists to this day.

As was mentioned above, the initialization problem domain can be broken down into four major areas:

- Standardization of data kept in service or national repositories
- Standardization of data used by automation systems, and the creation of a repository or repository system to house this data
- The creation of tools and processes to initialize systems using data from the repository, and
- Transition of the repository and tools to versions which are easily accessible and usable by warfighters (not engineers at some distant site) to provide a truly responsive capability to update initialization data as required.

We shall address each of these in turn, describing the state of ACSIS' progress in addressing them and recommending a general path forward for DoD, with probable implications for multinational initialization capability (IC).

Standardization or Synchronization of data kept in service or national repositories

Very early during the Army's digitization process, the CTSF team found that different systems used different 'authoritative' sources to initialize. For instance, the TRW team producing the FBCB2 database used the systems architecture produced by Program Executive Office—Command and Control Tactical (PEO-C3T), and simply created identifiers such as unit reference numbers (URNs) and host names and addressing data such as IP addresses. These identifiers and addresses were unknown to other entities such as combatant commands—which were busy assigning their own data—until the release of the FBCB2 database to the Army. The Combat Service Support Control System (CSSCS, since renamed) used unit identification codes (UICs) from the Standard Army Management Information System (STAMIS); the Global Command and Control System (GCCS) used UICs from the DoD Global Status of Resources and Training (GSORTS). Obviously, conflicts were inevitable, and were quite common; and there was little or no agreement on what constituted an 'authoritative' source, let alone a quality-control or feedback process for these sources.

The reason for this plethora of sources is perhaps obvious: **Until the advent of ACSIS, no one source contained all the data required to initialize our system of systems.** GSORTS contains UICs for company-level and larger units, and two name formats for each; the PEO-C3T systems architecture contains yet another name format, but identifies entities down to individual systems; the FBCB2 database modifies the name format and adds URNs and IP addresses among other data items. The situation is exacerbated because release dates for source data are not synchronized: One system may use GSORTS data current as of one date, while another system may have data from another date. It is hardly surprising that disparate systems often do not interoperate well.

The problem was, and is, further complicated by the lack of normalized data models in many data sources. Quite simply, these sources lack a database key—an unchanging identifier for any given entity in the database. Hence, deliveries from these sources, commonly including tens of thousands of records, may include dozens or hundreds of records that cannot be mapped to those in previous deliveries or to the contents of other repositories. The analyst or engineer is forced to rely on his or her military judgment to estimate which entities are being described in the new delivery—a slow, imprecise, and expensive effort that must be repeated with each new delivery. These deliveries can be up to daily in frequency, further exacerbating the problem.

Recently, DoD has initiated an effort to assign a force structure ID (FSID) to each entity in its databases, a development which promises to make the current process much more efficient. Although definitive guidance is still pending as of this writing, previous research and publications by the Army Staff[3] and Army Research Laboratory recommend a 64-bit identifier consisting of a 32-bit "prefix" identifying the assigner of an FSID, and a 32-bit suffix identifying the entity itself. Quite obviously, this scheme has the capacity to assign a unique identifier to every entity (person, unit, system, network) in DoD with room to spare, but guidance is so far lacking on coordination of these assignments between sources: For instance, there appears to be no guarantee or even recommendation that GSORTS will assign the same identifier suffix to a unit as will the Army's or the Marine Corps' force structure managers. Hence, for the foreseeable future, initialization engineers will be forced to rely on a partial solution to database keys, manually mapping entries between sources.

The fact that a source is 'official' or 'authoritative' is no guarantee that its data will be of high quality—in fact, the contrary situation often seems to exist[4], with no explicit quality controls on inputs. Finally, initialization engineers until recently lacked QC or feedback processes to these sources, so a key component of high-quality initialization data was lacking.

Although, given their magnitude, complete implementation of a solution to these problems will be a lengthy and difficult process, ACSIS has shown the way to do so through a number of initiatives:

- Initial development of interfaces to authoritative sources, including feedback processes, such that data from these sources can be synchronized with that from other sources and with existing products
- Development of an automated process to quality-check (QC) data and import it to an IC database
- Implementation of data-integrity features in the ACSIS database itself, focusing on the assignment of a unique identifier to each entity
- Development of an intellectual framework to qualify authoritative sources

ACSIS has developed interfaces of varying levels of sophistication with several authoritative data sources, including GSORTS; the Army's Force Management Support Agency (USAFMSA), which is responsible for the structure of Army tactical organizations; and the PEO-C3T systems architecture (SA), which provides high-definition information on the architecture of selected "digital" Army units. Development of these interfaces is of course crucial to the development of an overall initialization system, and perhaps not surprisingly has proved to be quite difficult in the current organizational environment. No Army or DoD agency has a clear picture of the entire initialization problem, and all are concerned with the daily challenges of managing force structure or supporting operational units, so resourcing interface development is a severe challenge. While the ACSIS team has been fortunate to have the active support of the Program Executive Officer for C3T, who has mandated development of interfaces to ACSIS from within PEO-C3T organizations, this remains a difficult challenge overall.

Because of ACSIS' birth in the challenges of Army digitization, the best-developed interface to date is that to the PEO-C3T systems architecture (SA). The PEO-C3T SA is a database and accompanying set of diagrams describing a reasonably detailed systems architecture for each "digitized" Army unit: For instance, it describes the physical location of each host, router, switch, and radio, and provides a general description of the networks of which each is a member. (It does not, however, provide detailed network and resource-allocation information, such as IP addressing or radio net assignments. How ACSIS makes these assignments is described in a later section.) The ACSIS interface to the PEO-C3T SA works by importing the SA database into an intermediate "staging" database where quality checks are performed and feedback generated. When the SA database passes quality checks it is imported into the ACSIS. Currently, however, the feedback process consists of text-based feedback from the ACSIS team to the PEO-C3T architects; it is evident that a fully-automated feedback process would make the process much more efficient.

Interfaces to GSORTS and USAFMSA are accomplished through the ACSIS Authoritative Data Source Update Process or AADSUP, a tool that downloads and quality-checks data from each source, and then in turn exports it to ACSIS. The development of AADSUP's code-based downloading and QC has proven to be a particular challenge because of the disparate formats in each source, the lack of database keys as previously described, and the fact that force structure data in these sources is only detailed to company level. To build initialization data for systems, data must be gained or created down to individual system level. AADSUP provides this capability through rules-based logic that parses company-level force structure data and segregates it into entities down to the squad or team level—the lowest level at which automation systems currently reside in our tactical units. Further development of AADSUP to address individual-soldier-level automation systems is a possibility that may be addressed in future versions of ACSIS.

Synchronization of the data received from these sources is performed both before and while the data resides in the staging database, and is the subject of extensive effort due to the fact that data from many 'authoritative' sources are already in fielded products such as the AFATDS Joint Master Unit List (JMUL). Hence, every effort is made to synchronize data from various sources while making minimal changes to fielded products. (This is the reason the team found it necessary to synchronize data from sources like the FBCB2 database, which would otherwise not be considered 'authoritative.') Software tools such as AADSUP are used to compare the contents of data sources, but inevitably dozens or hundreds of records are culled out and must be directly checked by the analyst. An example of the types of data which must be synchronized between various 'authoritative' sources is shown in Figure 2. In the figure, a few examples include operations where the Force Integration Office (FIO) Operational Facility (OPFAC) Name is compared to the FBCB2 Database Role Abbreviation, and the

FIO Organization Name (ORG NAME) is compared to the same data in the ACSIS–kept Modification Table of Organization and Equipment (MTOE). The numbers by each entry in the table correspond to the number of records in the release of the source that was the subject of the synchronization effort. In each case, the reader should remember that data synchronization is a process that must be repeated for each new delivery of data from an authoritative source.

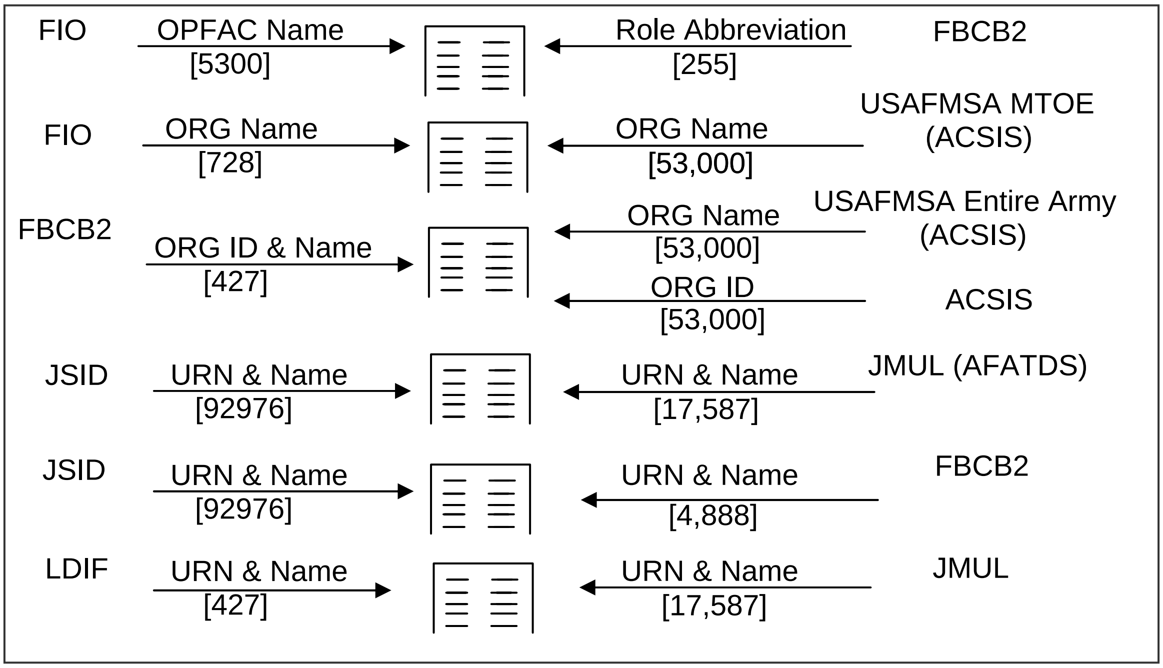

Figure 2. Examples of Data Synchronization Challenges.

In addition to synchronizing the data, the ACSIS team runs a number of analyses to ensure that the data provided is complete and accurate. For example, 'delta' reports highlight changes between a version and its successor, comparison analyses confirm the common elements among different data sources' products are comparable, and inventory counts confirm that data elements are accounted for from their source through the IC process.

A promising concept under development by the ACSIS team is the concept of a maturity model for authoritative data sources[5] based on the presence and repeatability of a review and authentication process for the data from a given source. Under the current concept, maturity levels would range from zero (no review or authentication process) to 4 (repeatable, subject-matter-expert based review process for all products). The goal is for the authoritative sources to provide analysis information about their data in addition to the data itself and to quantify how and whether the data was reviewed by subject matter experts and consumers. While clearly in its early stages, this concept promises to facilitate the continuous improvement of data sources into widely accepted and understood sources of data.

Standardization of data used by automation systems, and the creation of a repository or repository system to house this data

The ACSIS team found during its work that data from repositories is often closely tied to the command and control systems that originally used the data. For instance, data from the AFATDS JMUL includes system name formats readable by AFATDS systems only; and FBCB2 databases of course contain name formats for FBCB2 only. This in turn leads to widely varying database schemata and software implementations, which not only name entities differently but often handle them in different ways. For instance, the DoD's Common Operational Picture (COP) in the Global Command and Control System (GCCS) depicts battlespace objects by their positions over time (tracks), while the Army's Common Tactical Picture (CTP) depicts an object by a single icon which moves on the display according to its actual position at any instant..

Not surprisingly, these data schema choices are related to the message formats in use by the systems in question. While this would seem to promise some form of standardization, in fact the reverse is largely true because of the large number of message formats in use. Even a very brief survey of message formats in use by DoD reveals the case: US Message Text Format (USMTF) 1993 uses a different name format in message headers than do later versions; Variable Message Format (VMF) messages blessedly use the same format as later USMTF versions, but unfortunately do not use the same format as the popular Over-the-Horizon Gold (OTH-G) messages; and the new extensible markup language (XML)-based messaging formats do not specify a rigorous naming convention at all.

It is worth noting that the use of XML can reduce the problem of sending various name formats by providing numerous easily-available schemata, but does not solve the basic problem. The multiplicity of schemata is itself confusing, and the fact that an XML tag can describe, say, a "name" (Unit name? Host name? Billet name?) leaves no assurance that the name will be parsable by the system software, related to the contents of its database, or distinct or understandable to the user. (We of course also recognize the various parameters such as processor power and bandwidth availability which have prompted the creation and use of various message formats, and do not believe these standards can be merged or downselected in the foreseeable future.)

All this, of course, leads to greatly increased cost both in populating and maintaining system databases and data sources, and in creating and maintaining software interfaces between systems. It has been our experience at the CTSF that even demonstrated adherence to a single messaging or graphics visualization standard is no guarantee that systems will interoperate without at least some modification to their software as the results of interoperability testing.

While we recognize the problems associated with the lack of standardization, the ACSIS team is obviously not in a position to dictate changes to system software or to the choice of messaging standards. Hence, ACSIS is a very customer-focused product, which houses data, and creates and maintains tools, to initialize a wide variety of systems. Although we are unaware of any studies of the subject, we surmise it may in fact be less expensive in the near term to maintain such a sophisticated data repository than it would be to make extensive changes to the software of numerous command-and-control and communications systems.

Hence, we believe ACSIS is a prototype for IC in DoD and perhaps in the multinational arena. While ACSIS currently can initialize Army and Marine Corps systems, we have no reason to believe the data model and tools could not be easily modified to work with other services' or other nations' systems. We believe the key to developing this multi-service (in US parlance, "Joint") capability is a multi-service requirement for IC, from which the whole problem of IC—not only the repository, but resourcing and data standardization—can be addressed in a systematic fashion.

The key to ACSIS' flexibility as a common repository is a robust yet comprehensible data model and a set of common tool interfaces through which systems can be initialized with data from the repository. We shall now turn to an examination of the data model.

The ACSIS Data Model

Because the demands of customers could not—and still cannot—be entirely foreseen, the ACSIS database designers created and maintained the data model in classic third-normal form to enforce key and relational integrity even through the process of various database manipulations. This allows the team to be confident in the quality of database outputs while allowing great flexibility in the format and content of products built through the tool set. An overview of the ACSIS data model is at Figure 3.

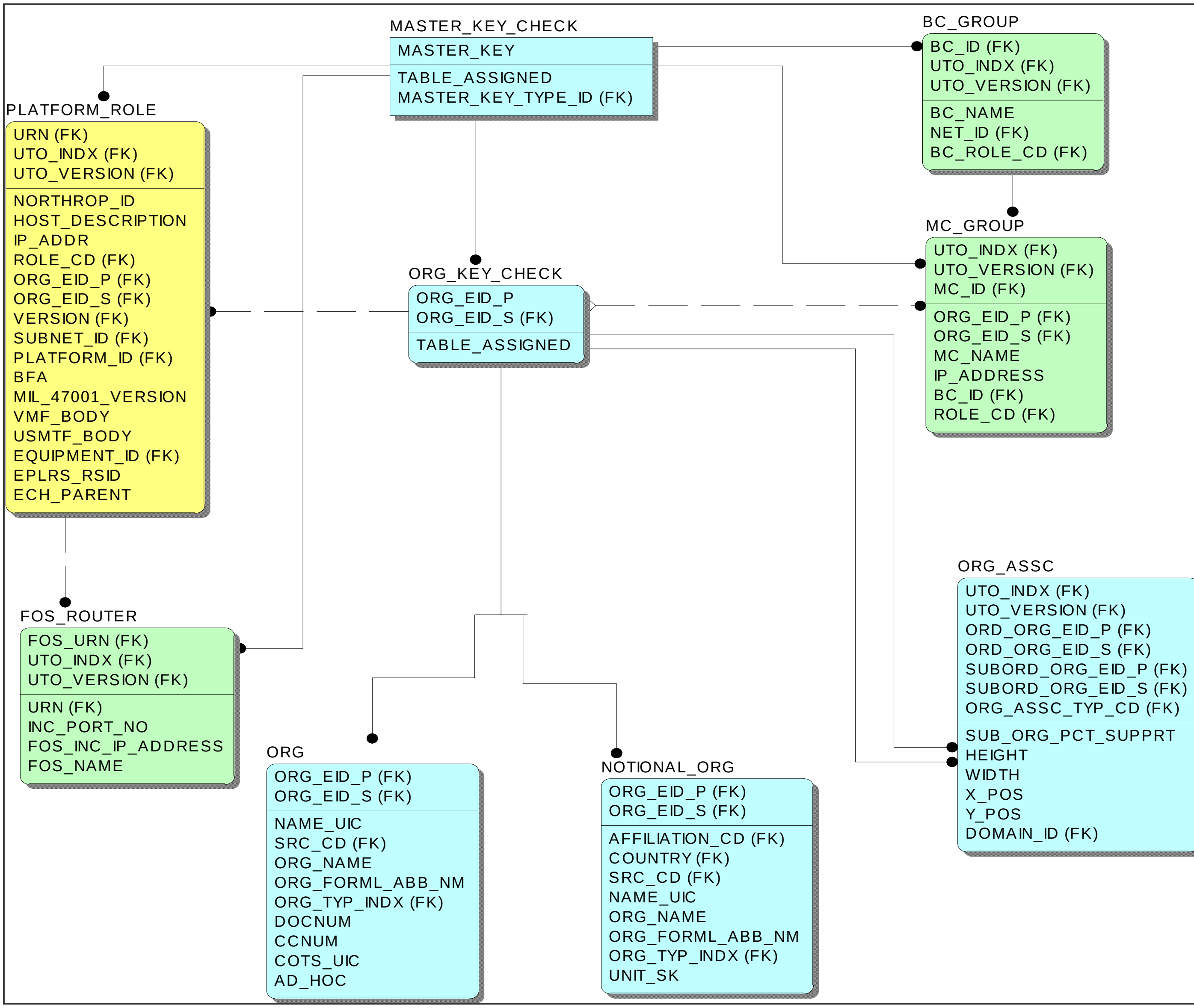

Figure 3. ACSIS Data Model Overview.

Of particular note in the overview is the use of master keys and organization keys, allowing flexible assignment and rapid indexing of database keys throughout the database. In this manner, keys for different types of entities, or even different keys for the same general types of entities, can be maintained without violating data integrity.

A view of the ACSIS organization and organization association data is shown at Figure 4.

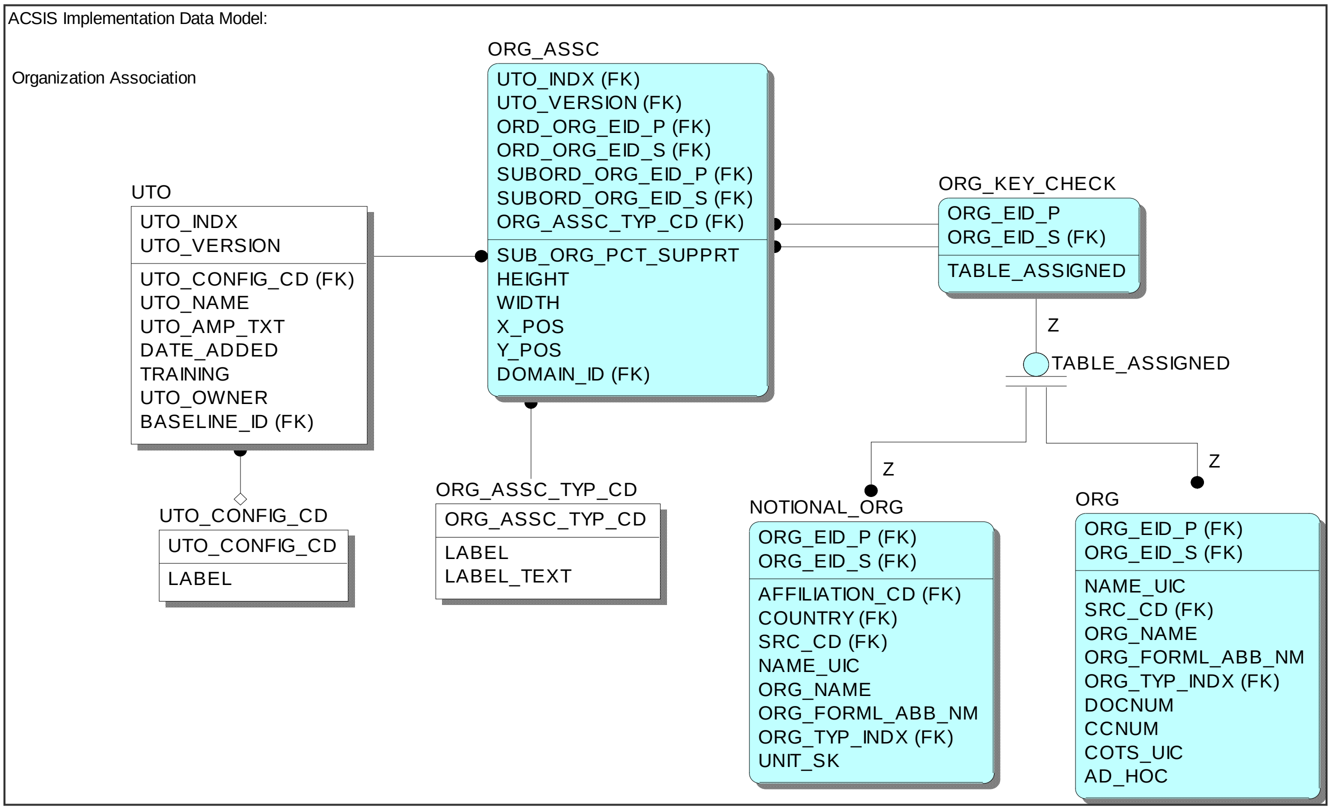

Figure 4. ACSIS Master Key, Organization and Organization Association Data.

The organization association (ORG_ASSC) table maintains the troop list and task organization for the unit of interest to the database and production teams. By convention, unit task organization (UTO) 1.0 contains all data in the database, while versions 2.0 and higher contains specific task organizations for customer units requiring initialization data. The use of major (for example, 2.0) and minor (2.1, 2.2) revisions to task organizations allow customer units to be easily identified while making successive changes to their troop list and task organization. The table also contains entries to easily allow description of the type of association between organizations (such as Attached or Operational Command) and the percent support given to a supported unit by a given organization.

Also of note are the use of organization and notional organization tables, thus allowing the database and tools to differentiate between real organizations and those that might exist only in simulations. This allows the production team to rapidly and confidently build product sets for both live and simulation environments without the possibility of confusion over the nature of a given unit. Of course, it is easily possible to include real units in simulations, such as is commonly done for US Army command post exercises like the Battle Command Training Program (BCTP).

The organization table itself allows for the use of both formally organized and *ad hoc* units such as the Forward Logistics Elements (FLEs) commonly used by the US Army.

As the reader will infer, the nature of *ad hoc* units forced the database team to make nullable many fields in this table such as Unit Identification Code (Name_UIC), in turn forcing reliance on the Organization Enterprise Identifier (Org_EID). This reliance once again reinforces the utility of the nascent Force Structure Identifier concept in DoD.

Figure 5 depicts the handling of individual systems ("Platforms") and their associations to broadcast and multicast groups.

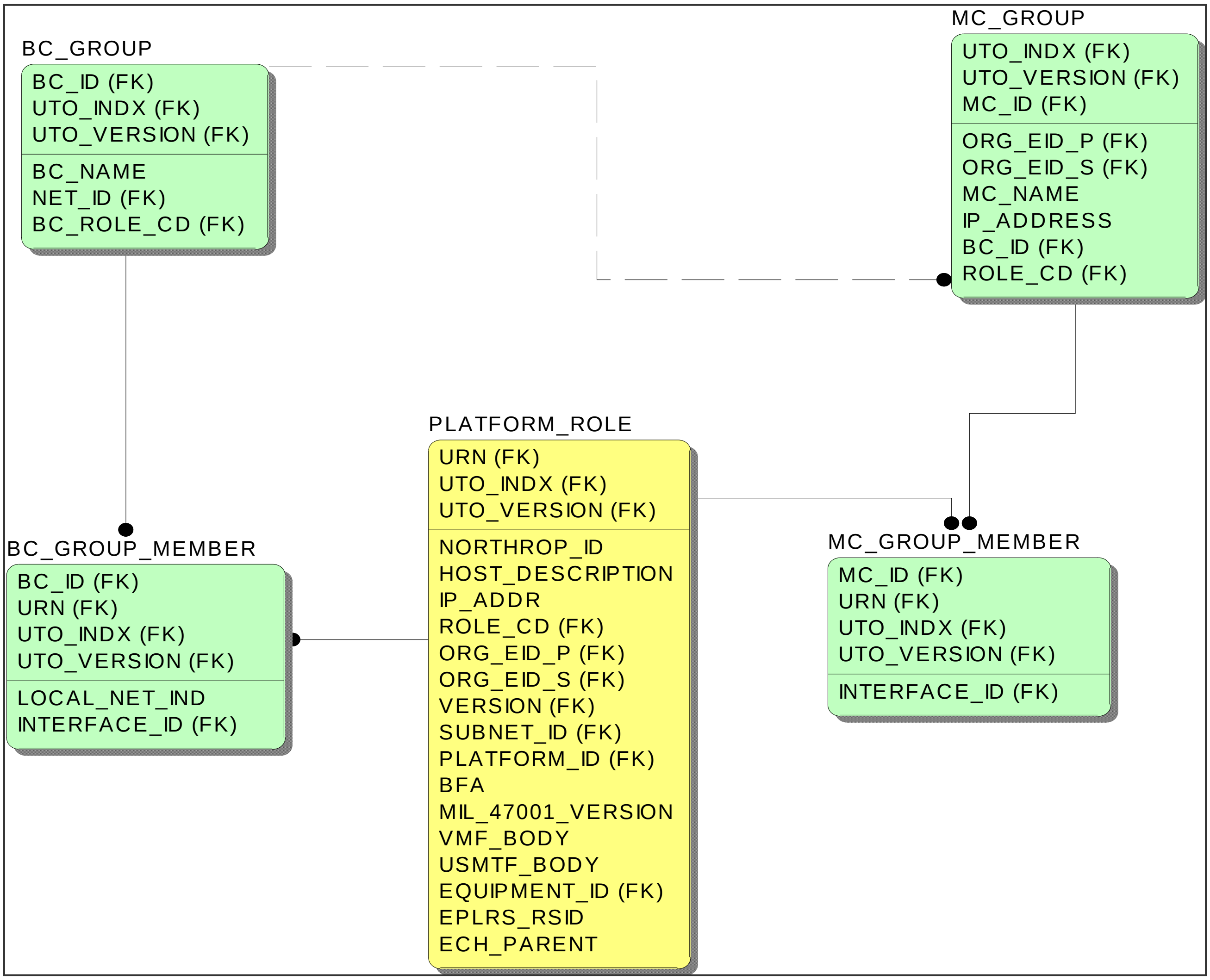

Figure 5. Platforms and Their Group Associations

That platform_role table and its associated group tables illustrate once again the customer-focused nature of the ACSIS data model. Note, for instance, the communications-system-specific fields in the platform_role table such as MIL_47001_VERSION (the version of VMF messages the platform supports) and EPLRS_RSID (the unique identifier for an Enhanced Position Location Reporting System (EPLRS) radio associated with the platform). While we do not foresee a technical problem associated with continuing modifications to the data model in response to new "customer" systems, there is quite clearly a significant expense associated with populating and maintaining an ever-expanding data model. While we would like to *derive* (rather than warehouse and maintain) the contents of many such fields in system initialization products, the hard truth is that algorithms to perform these operations are by no means easy to construct, are significantly expensive to code, and therefore place a significant resource demand on the team building the initialization tool. While maintaining the third-normal characteristics of the data model in the face of continuing expansion is also not an insignificant effort, it is probably less costly in the near term than placing an unexpected burden on a system team to derive its data from the existing model.

Materiel data tables are shown in Figure 6.

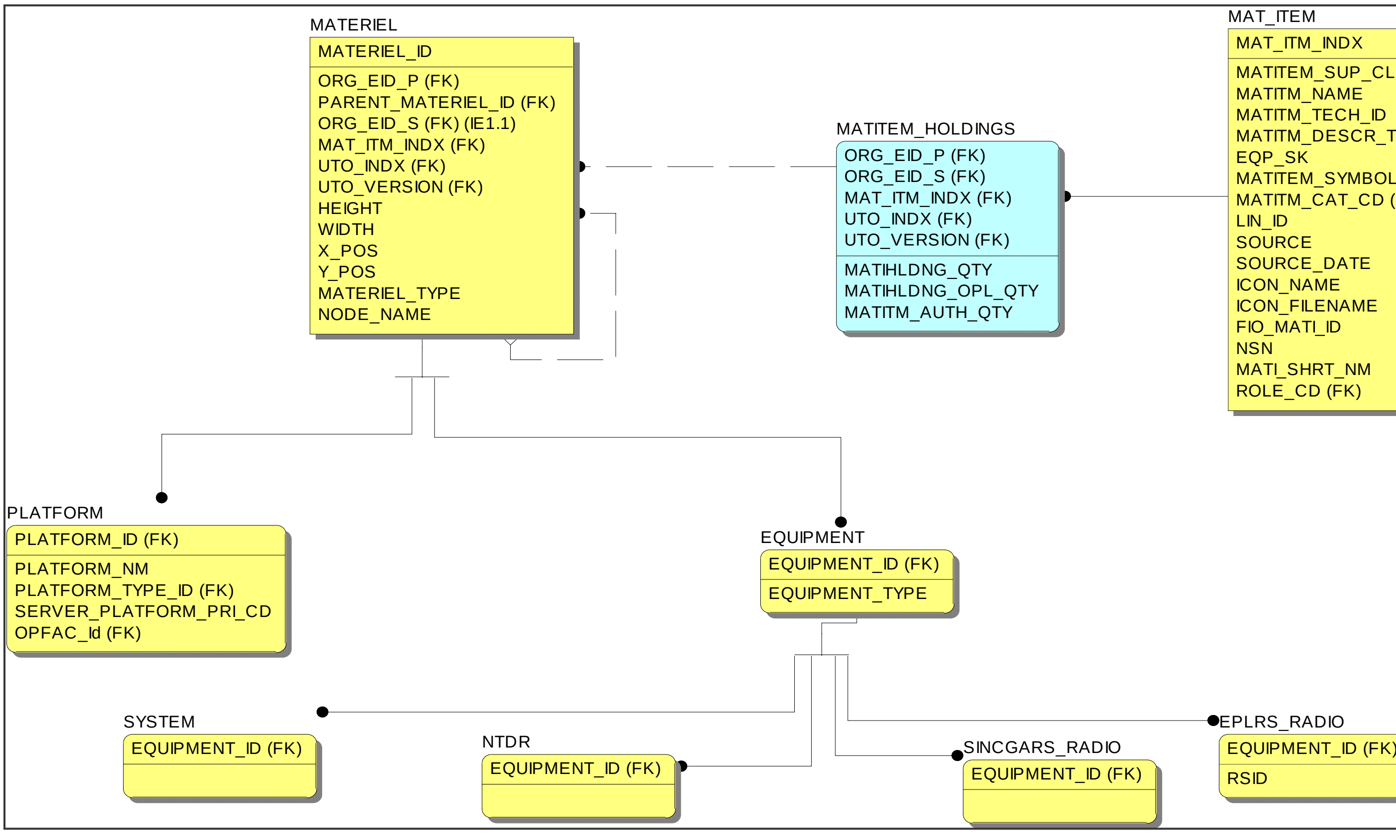

Figure 6. ACSIS Materiel Tables.

The contents of the materiel tables are currently based on US Army logistics data items but could be readily modified to accommodate different fields according to a customer's needs.

ACSIS Personnel tables are shown in Figure 7.

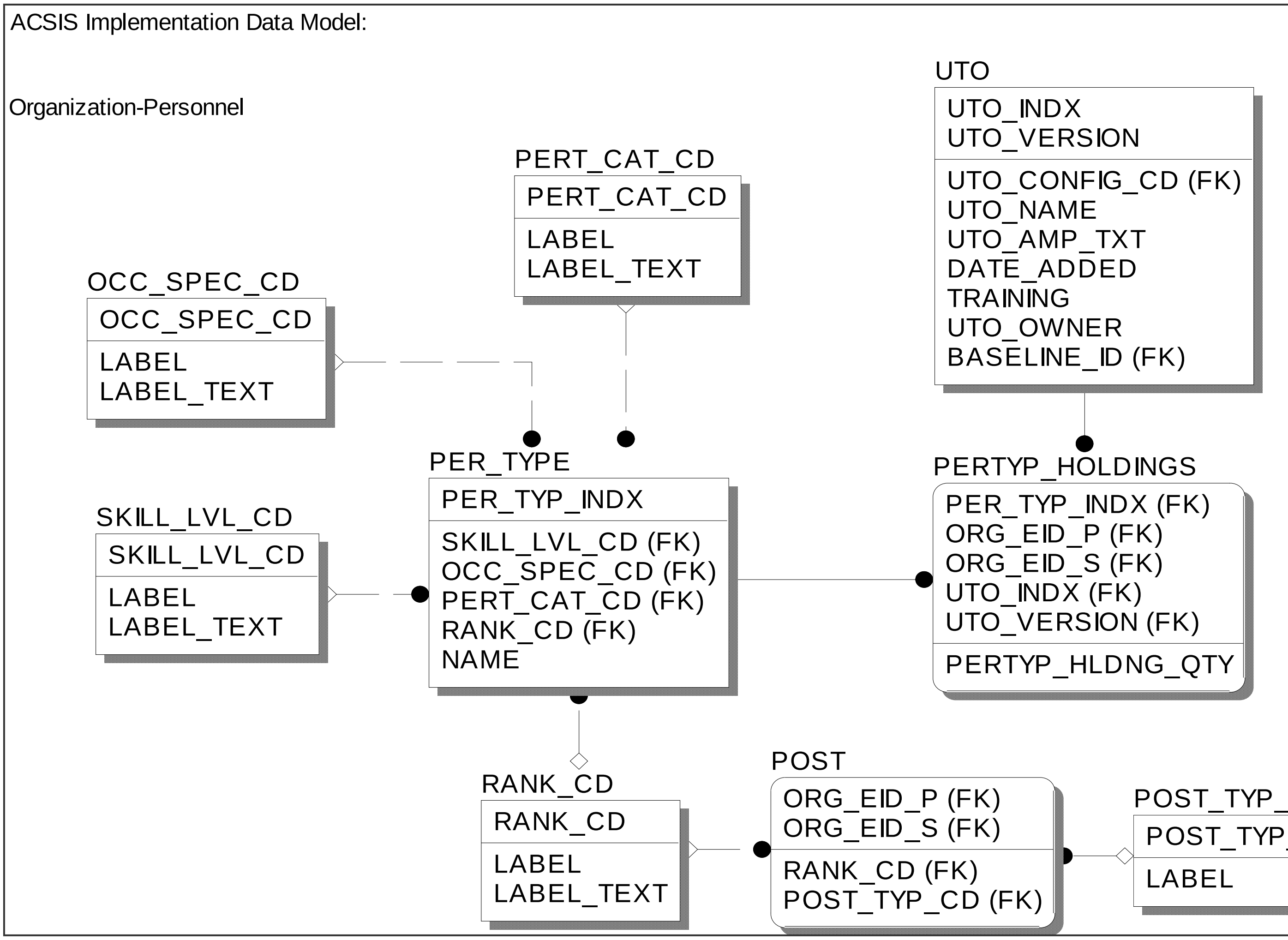

Figure 7. ACSIS Personnel Tables.

Currently, personnel tables in the ACSIS are only populated for personnel holdings in notional units in the simulation environment. The reader will easily infer that the data model focuses on the personnel *holdings* of subject units and is not yet fully developed in terms of *personal* identifiers such as the US Social Security Number (SSN)--but once again could be easily modified to accommodate these identifiers. An interesting possibility in terms of personnel is that of identifying an individual by a unique identifier other than the SSN, which is seen by many as sensitive information which should not be disseminated. Current DA and DoD thinking on the FSID indicates that this identifier could be used in just such a way[6] in this table.

Two depictions of networks and communications tables are at figures 8 and 9.

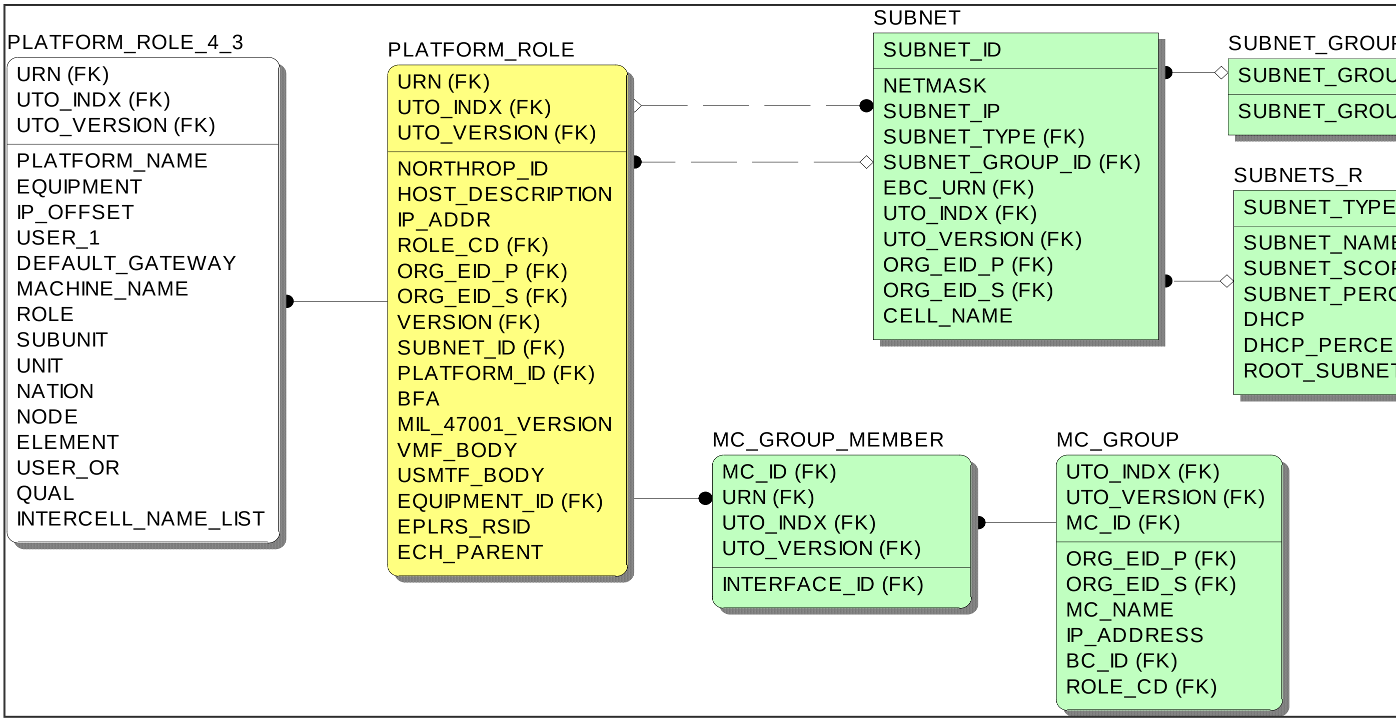

Figure 8. ACSIS Networks and Communications, Part I.

Figure 8 illustrates the overall structure of the networks and communications data structure, illustrating the importance of network structures (such as subnet allocations) and of the use of multicast in our tactical architectures. This in turn once again illustrates the customer-focused nature of ACSIS: While multicast group memberships might possibly be derived from organization associations ("task organizations") based on a known rule set, the ubiquity of multicast in US architectures prompted the database team to simply include multicast group memberships directly in the data model.

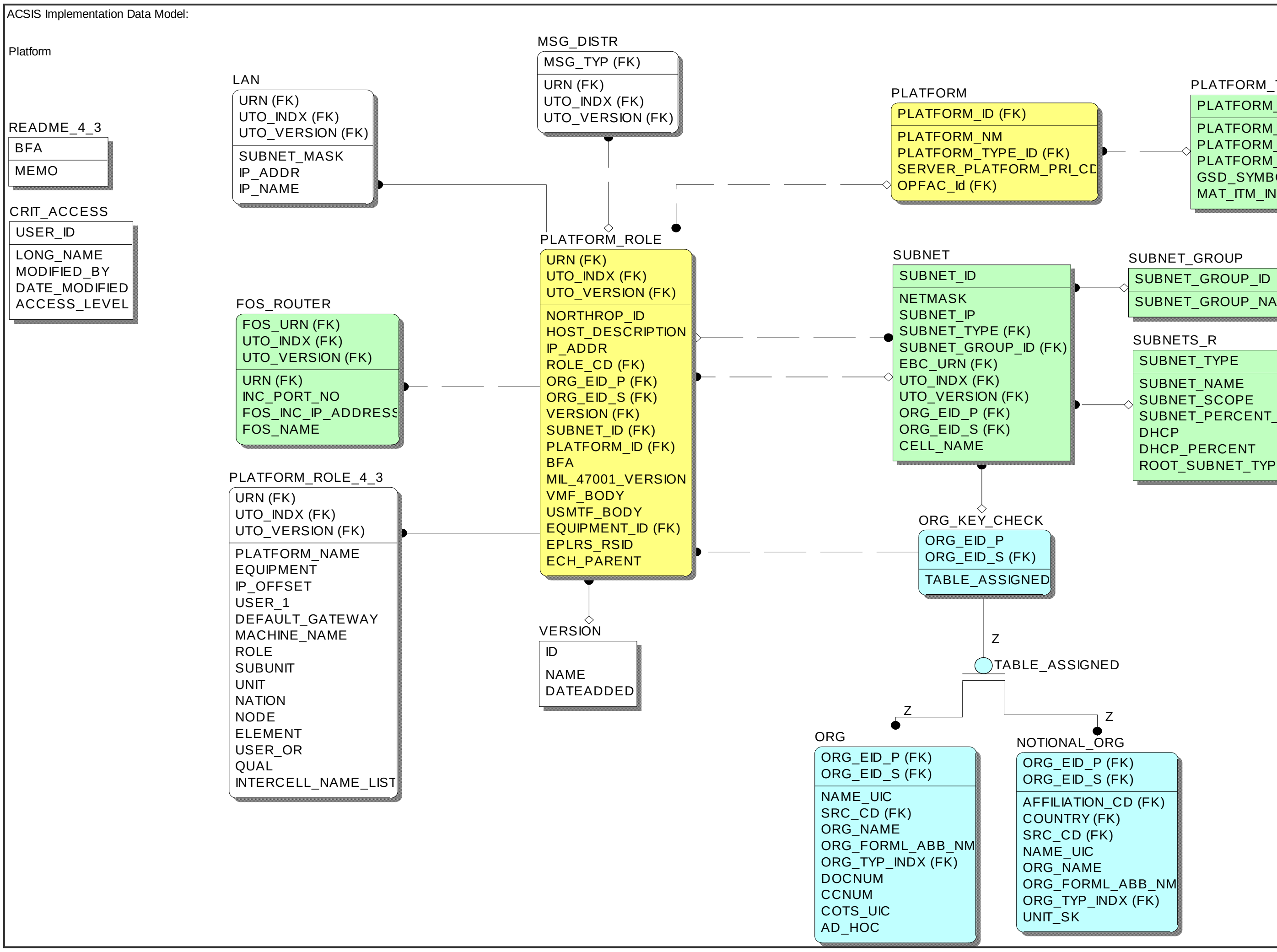

Figure 9. ACSIS Networks and Communications, Part II.

Figure 9 illustrates the association of platform, and subnets and groups, to their parent organizations and task organizations. It illustrates yet again the customer focus of ACSIS with the FOS_ROUTER, MSG_DISTR, and LAN tables directly focused on the FBCB2 destination system.

The current ACSIS architecture is focused on only two production database instances, one each at Forts Hood and Monmouth, with a replication scheme to allow currency and immediate backup. While clearly rudimentary in terms of an objective architecture, it is giving the team solid experience in defining practical database replication schemes in preparation for future work. Of particular note, however, in the multinational context is the fact that ACSIS is resident only on the US Secret Internet Protocol Routed Network (SIPRNet), so is not yet accessible to coalition partners or even to unclassified users.

The ACSIS team is addressing this issue early by planning and preparing for a multi-level security implementation using the Oracle database management system (DBMS) and the Trusted Solaris operating system. (Currently, ACSIS runs on the Oracle 9i DBMS and Solaris 9 because of their reliability, easy management of large architectures, and scalability. The team is considering whether to ‘regress’ to the Trusted Solaris, based on Solaris 8, or to wait for certification of Solaris 10 and its trusted features. Of particular note is that Oracle is the only DBMS of which the team is aware which incorporates EU Common Criteria certified multi-level security features, reinforcing the team’s original choice of a DBMS.) Clearly, while a multi-level secure database in no way addresses all the numerous issues related to multi-level security in a multinational network, it does offer a promising start in sharing data internationally in the face of national-level security concerns.

The team foresees the eventual architecture as that of individual database instances with connecting tool suites fielded down to and including brigade or unit of action (UA) level, with a replication scheme in effect not only among tactical nodes but between the tactical and the sustaining base environment; the latter would include several instances of a Network-Centric Data Management Center of NDMC. A schematic diagram of this arrangement is shown at Figure 10.

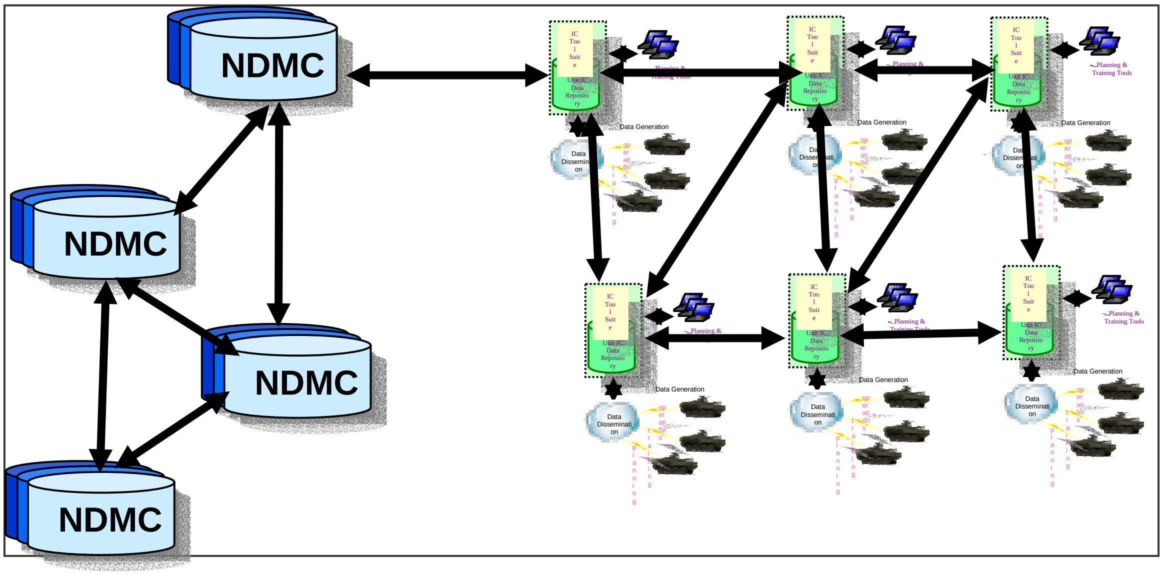

Figure 10. Conceptual Representation of Eventual Distributed Database Architecture.

As previously mentioned, currently the ACSIS and its associated Web-Enabled Address Book or WAB are only accessible over the US-only SIPRNet. This brings to the fore the looming policy issues of data sharing with coalition partners and what can be shared with some partners but not others. While in general not technically difficult to the database designer or administrator equipped with Oracle, issues such as these obviously present severe organizational challenges to the team and its sponsors.

Tools and processes to initialize systems using data from the repository

Clearly, the best data repository and sources are of little use if the data cannot be quickly and reliably used to initialize the destination systems. As was mentioned in the introduction to this paper, one of the original—and for many systems still-existing—means to build initialization products was quite literally by hand, manually populating database tables and then testing the product on its host system by performing representative system functions in a laboratory simulation of the live environment. Again somewhat obviously, this process suffers from several severe shortcomings. First, manual population of large database tables is very slow and error-prone, defeating the promise of automation to rapidly deliver repeatable, reliable results. Second, it offers no assurance that database products for different systems in a given architecture will be consistent with one another. Finally, direct testing of the product through operator manipulations is also slow, and cannot reasonably replicate modern large architectures which may have literally thousands of nodes.[7]

The ACSIS team tackled these problems by building a tool set interfacing to the ACSIS database to build initialization products for destination systems. As might be inferred from the CTSF's lead role in integrating the Army Battle Command System or ABCS, the first tool set for ACSIS initialized ABCS version 6.2, then in use by the Army's "digitized" units, and the accompanying simulation systems. Subsequent builds of the tool created the capability to initialize so-called "legacy" or "4.3" ABCS systems, ABCS 6.3 Delta, and the current ABCS 6.4, with associated tools to initialize simulations accompanying each tool delivery. The tool suite is built in modular fashion to allow easy upgrades to any desired capability without serious effects on other tools.

Of particular interest in the tool arena is the rule-based build of network groups such as subnets, multicast groups, and radio nets. This area was a particular challenge for the team because of the lack of documentation for and the relative opacity of the software logic in the destination systems. The CTSF engineering staff knows by experience, for instance, the practical limits on subnet and information-management group sizes for various software items, but system documentation to confirm or refute this experience was—and is—often lacking. As a result, the team spent seemingly countless hours interviewing engineers and soldiers and distilling their experience into a practical rule set for a given software item and military unit type, then many more hours designing and coding tool software to produce products which adhered to these rules. In this way, the tool is able to produce a consistent and functional set of groups for any relevant software version and virtually any military task organization which might use the software. This rules-gathering effort is by no means complete and we expect it will continue for the foreseeable future as more units and systems turn to ACSIS for initialization products.

Because of programmatic considerations, the ACSIS team did not build all of the tools to initialize ABCS systems. PM-FBCB2 and Product Manager (PdM) Tactical Internet Management System (TIMS) continue to build the initialization products for their systems; these products also serve to initialize the EPLRS and Single Channel Ground-Air Radio System or SINCGARS, and to populate subnets and multicast groups in the "digitized" units. The astute reader will have noted that this initialization data is part of the ACSIS data model itself, leading to the correct conclusion that these products are imported into ACSIS before completing the build of a product set for these architectures.

The concept of importing some of ACSIS' contents from the initialization processes themselves offers both benefits and drawbacks to the project. The benefits include some measure of workload-sharing in terms of creating products, and a small gain in performance for the fielded product set since PM-FBCB2 has the most experience at efficiently initializing and managing digital radio nets in the TI. The drawbacks include complexity and a reliance on a "subcontractor" which may not be willing to share its product and process specifications and standards. While these latter considerations are by no means insignificant, it is quite clear that the small ACSIS team cannot create tools to initialize the dozens or hundreds of disparate systems in DoD, let alone those of coalition partners. Hence, the team has chosen to publish a number of application program interfaces (APIs) against which systems can program their tools, accepting the complexity in some cases of importing data built from these less-than-transparent processes. The team estimates that this type of interface will grow in importance as the full scope of the initialization challenge is determined.

An associated challenge for the team was to create a validation tool for the data products themselves, giving the user a level of confidence in the finished product without the need for extensive testing on an actual architecture. While this process has not yet gained full acceptance and the CTSF still tests data products explicitly, the verified quality of ACSIS-built products on the CTSF test floor makes this prospect a strong possibility in the near future. Since explicit tests of data products on the CTSF test floor can take as long as five days and use up to a third of the CTSF's available test resources, it is clear that accelerating production rates make this validation process mandatory.

The ACSIS team is facilitating the acceptance of "validated" or "proven" products in advance of formal testing through the conduct of data review boards or DRBs. The DRB is an explicit and thorough review of a product set by all interested parties, ranging from the system architects to using units to representatives from the CTSF test floor and its software support teams. Typically, after the build of a product set the parties are given 2-3 days to analyze the product set and prepare lists of comments or concerns; then the review, a half- or full-day affair, is conducted to review and respond to these issues. Comments raised range from requested changes to system name formats to requests for the addition of one or more hosts to a unit architecture to questions about IP address allocation. Typically, because of the ACSIS tool set, the team is able to respond to requests for such changes in near real time—an enormous improvement over past processes. One note of caution in this picture is that some products, such as the FBCB2 database, are still hand-built and explicitly tested, making change requests a matter of weeks or even months—so the situation is not entirely bright.

Transition to a "warfighter" repository and tools

While, as we have noted, the ACSIS remains an engineers' and not a warfighters' tool, developments in PEO-C3T are accelerating the fielding of IC to the hands of the warfighter in advance of the true distributed database architecture of Figure 10. Chief among these is the Web-Enabled Address Book or WAB, which will achieve initial operational capability in early September 2004. The WAB consists of a web server interface for the ACSIS, through which using units can add hosts or make other changes to ACSIS, and in turn can create and download a new set of initialization products. The team is also considering creating a Microsoft Windows® System Update Server (SUS) to disseminate software changes to destination unit systems alongside the delivery of data products.

This step, while very positive, continues to depend on a unit having SIPRNet access to the CTSF in the continental United States. Especially for individual vehicle systems, such access is in many cases simply impossible. Hence, development of the distributed database architecture in Figure 10 remains a high priority for the team.

Path forward and Challenges

As ACSIS and IC have gained the notice and support of DoD's senior leaders, the future of IC has become much brighter than its past. Nevertheless, numerous challenges lie ahead for what remains a very small initiative by national and international standards. These challenges include:

- Requirements definition.
- Data synchronization and standardization with services other than the Army and Marine Corps
- Data standardization in our C4ISR and simulation systems
- Coordination with other initialization efforts

IC requirements definition, while seemingly straightforward at the conceptual level, is much less so in practice. This is because—at least in the US—requirements drive resourcing, so a precise definition of IC, including exactly which systems must be initialized (and with what data they must be initialized) is an essential step in requirements definition. Also, we concede that there remain other potential uses of IC data—such as for management information—with which IC's current practitioners are unfamiliar. The just-beginning DoD effort to initiate IC communities of interest (COIs) is a worthwhile effort to harness the knowledge of engineers and warfighters to develop an accurate and precise statement of what is required, though it will certainly take much more time than we would like to do so.

Data synchronization and standardization will remain severe challenges for IC as it proceeds into the multi-service arena. The ACSIS team has recently accepted the task to produce initialization products for US Marine Corps as well as for Army systems, and its initial work in this area—not surprisingly—indicates a data synchronization challenge among numerous new sources of data from which the ACSIS repository must draw. The team expects to encounter similar situations as it expands its work to other services.

In a similar vein, data standardization of our systems remains a challenge that is largely unconquered. The US Army's efforts in this area have focused on the use of

XML, which as described previously eases the problem only partially; and it is not clear to the ACSIS team whether DoD-level efforts will facilitate any standardization of data between our systems.

Finally, coordination between ACSIS and other initialization efforts will be a challenge as well. The team's mentors and contacts in DoD are unaware of any other efforts in the US which would perform a function similar to ACSIS, but there are numerous potential customers for such a capability including the Joint Forces Command's Rapid Distributed Database Development (RD3) effort which seeks to initialize C4ISR and simulation systems in our combat training centers. While we are hopeful that the COI effort will uncover many such opportunities for collaboration, it is much too early in the COI effort to be sure these will bear fruit.

The future of multinational IC is much less clear even than this due to the fledgling nature of international C4ISR interoperability efforts. (Even the term "C4ISR" is an American one and is not to our knowledge internationally agreed upon; we note that different countries use C2 or CC (Command and Control), C3 (C2 plus Communications), or other abbreviations.) Clearly, data synchronization will be an even more difficult problem than it is for US systems—although we are hopeful that standard data models like the Command and Control Information Exchange Data Model (C2IEDM) will provide a key to more rapidly map data items between systems and between repositories.

Probably much more difficult to address will be security considerations, as various nations may or may not be willing to share IC information with others. As mentioned earlier, this problem must not only be addressed at the level of the DBMS, but at the level of the network and quite possibly the communications system as well. Each of these layers is of course a significant cost factor as well as a policy challenge, since current policies require the installation of expensive multi-level security guard devices between networks of different classification levels. Though the course of multinational networks in Operation Iraqi Freedom (OIF) provides a hopeful example of cooperation, the provision of a "standard," permanent system and process will encounter many more regulatory and legal challenges as it proceeds towards operational capability.

Conclusion

For the ACSIS team, its work in the IC arena has often brought to mind the ancient myth of Sisyphus, who was doomed to painfully roll a boulder up a hill each day only to have it roll back down as he approached the top. The increasing awareness of IC within the senior leadership of the Army and DoD, however, has led us to the hope that Sisyphus will soon have collaborators who can help put the rock of IC at the top of the hill. We realize, however, that IC challenges will remain for the foreseeable future, as the larger and more numerous "rocks" of multi-service and multi-national IC await our work.

Notes

[1] Martin H. Weik, Ordnance Ballistic Research Laboratories, Aberdeen Proving Ground, MD, "The ENIAC Story," from US Army Research web site, http://www.ftp.arl.mil/~mike/comphist/eniac-story.html. Among other gems of initialization, Weik writes, "The principal purpose of the function tables, which actually were banks of switch-controlled resistor matrices, was the storage of the arbitrary functions call for by the problem."

[2] See, for instance, Colonel Trevor Dupuy, *A Genius for War: The German Army and General Staff, 1807-1945* (ISBN 0963869213), an extensive treatise of the German-initiated General Staff system which included extensive indoctrination in common language and terminology for staff functions. The current US Army Command and General Staff Officer Course, of which one of the authors of this paper is a graduate, strongly emphasizes such standardization in the US Army.

[3] Bruce Haberkamp, "Enterprise Identifers/Common Data Schema: Solutions for Data Interoperability," paper presented at 2001 Software Technology Conference, May 2001, presents this concept in detail. Hereinafter referred to as Haberkamp.

[4] For instance, Chairman of the Joint Chiefs of Staff Instruction (CJCSI) 3150.02A, *Global Status of Resources and Training System (GSORTS)*, Enclosure B (Reporting GSORTS Data) explains that all data fields in the Basic Identify Data Element (BIDE) (which identifies units in GSORTS) are under the control of the reporting unit. Perhaps most telling, none of the name fields in BIDE are even subject to a standard convention beyond the requirement for a maximum length and the use of alphanumeric or special characters.

[5] Frank J. Ponzio Jr., "Authoritative Data Source Framework White Paper," Symbolic Systems, Inc., November 2003, http://www.symbolic.com .

[6] Haberkamp, pp. 5-6.

[7] The CTSF, for instance, has in the authors' extensive experience never been able to create a test architecture larger than 200 nodes, while the network of a US Army digital division typically includes thousands of nodes. Higher unit networks, which of course include those of lower units like the divisions, are often much larger still.