

Homeland Security Incident Response Preplanning: Lessons Learned on the Importance of Initialization Data

Donald W. Jones, KITS, INC
Lieutenant General US Army, Retired
Vice President, Strategic Programs
djones@kits-inc.com

Randall V. Shane, KITS, INC
President and CTO
rshane@kits-inc.com

Abstract

To be capable of responding to any incident, regardless of its size, type, complexity, or location is a daunting task. The magnitude of this problem becomes staggering when all implications are considered. However, the efficiency with which the response is executed directly correlates to the amount of preplanning that has occurred. Additionally, to be effective, the preplanning phase should include the identification, registration, and training of available resources, along with an implemented system for rapidly activating and deploying the appropriate resources for a given incident.

One of the more frequently overlooked aspects in preplanning and deployment is the development of a rapid initialization¹ process. Additionally, the more stringent the security requirements, the more data needed, and the more preplanning required. The purpose of this paper is to address the importance of data engineering and its role in security, interoperability, and reliable information exchange, with the emphasis on initialization data.

Key points will be addressed in the form of “lessons learned”, drawing from the experience of the authors in their respective fields. Donald W. Jones is currently working as the Vice President of Strategic Programs for KITS, INC, and has served as the Senior Vice President for Chapter Services of the American Red Cross in addition to retiring from the United States Army with the permanent rank of Lieutenant General. Randall Shane works as the President and Chief Technology Officer for KITS, INC, and has worked as a senior systems engineer and chief architect on several System of Systems (SoS) enterprise level projects for the US Army.

1. Introduction

Industrial accidents, natural disasters, terrorist attacks, and planned public events, are examples of incidents

¹ For the purposes of this paper, initialization data is defined as the common set of data that must be initially loaded in each system in a given architecture as the primary enabler of interoperability and reliable information exchange.

requiring quick, appropriate, well-organized responses to minimize loss of life and property.

An obvious requirement for any system of this importance is a robust, scalable, secure, and flexible communications infrastructure as a key enabler of reliable information exchange. Furthermore, this infrastructure must be available under the most extreme circumstances such as the terrorist attacks on 9/11, 2001, the tsunami that devastated Indonesia in 2004, and hurricanes Katrina and Rita in 2005.

This system must be designed such that the complexities are transparent to the end users and flexible enough to allow for the deployment of resources needed to maximize the effectiveness of a given response. To demonstrate the importance of initialization data as an enabler of such a system, this paper draws an analogy between a possible command and control system designed for Homeland Security, and the Army's enterprise level C4ISR systems (Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance). The complete architecture, to include Enterprise Service Busses (ESB), and Service Oriented Architectures (SOA) is beyond the scope of this paper.

2. Level of Sophistication Required

Under Homeland Security Presidential Directive (HSPD)-5, the President of the US mandated the development of the National Incident Management System (NIMS)², which provides a comprehensive template for managing incidents. The National Response Plan (NRP)³, built from the NIMS template, encompasses not only the prevention of, but also the response to, and recovery from major disasters.

There are now policies, procedures, and processes in place that provide a coherent doctrinal framework across all jurisdictional, organizational, and operational boundaries for incident management, regardless of the cause, size, or complexity of the incident. However,

² See the National Incident Management System Integration Center web site at <http://www.fema.gov/emergency/nims/index.shtm>

³ See the National Response Plan web site at: http://www.dhs.gov/xprepresp/committees/editorial_0566.shtm

having a solid framework in place is no substitute for a sophisticated implementation.

For example, the Incident Command System (ICS)⁴ National Response System (NRS) – Concept of Response states that the Federal On-Scene Coordinator (FOSC) is “responsible for immediately collecting pertinent facts about the [incident]. . .” However, it does not state how this information dissemination will occur. Does the FOSC receive a phone call, an email, watch the news, or log on to an incident site and read situational awareness (SA) updates? Who contacts the FOSC with this information, or who does the FOSC contact to receive the information? How would the FOSC, or authorities from any group receive the necessary information for making decisions? Would decision analysis algorithms be needed to ensure consistent responses across agencies?

It is easy to see how the measure of success for any emergency response system is dependent on effective communications, and therefore, on the sophistication of implementation⁵. As will be explained in this paper, there must be a comprehensive and integrated solution that maximizes the effectiveness of any agency’s incident response by enabling system interoperability and reliable information exchange. Furthermore, the discussion will center on the importance of preplanning, and the development of an initialization system, explaining how it is critical to the successful execution of any incident response system.

3. Lessons Learned – Enterprise Problems and Enterprise Solutions

The United States Department of Defense (DoD) is probably the largest organization to ever attempt large-scale interoperability. Therefore, it makes sense that their lessons learned should be examined closely for any proposed solution for Homeland Security. Within the DoD, the Army is the largest organization, and is also the organization that has many lessons from which to draw.

Several decades ago, the Army began developing the concept of integrated system communications as a force multiplier. Today, these integrated systems are known collectively as the Army Battle Command System (ABCS)⁶. There are numerous systems inside this system of systems (SoS) architecture, each having a functional area of responsibility. The system within ABCS that has proven one of the most effective, and whose value has

⁴ See NIMS Integration Center Discussion on ICS at: <http://www.fema.gov/news/newsrelease.fema?id=15556>
<http://www.training.fema.gov/EMIWeb/IS/ICSResource/index.htm>

⁵ This discussion is not intended to detract from the necessity for, or value of ICS and the concept of response. In fact, it would be illogical to attempt an implementation without a defined framework and the two are considered complementary.

⁶ <http://www.fas.org/man/dod-101/sys/land/abcs.htm>

been proven in combat, is the Force XXI Battle Command Brigade and Below (FBCB2) system.

FBCB2 is a command and control system that provides both horizontal, and vertical communications, command and control (C2) messaging, and is best known and appreciated for its ability to provide position reports on all systems equipped with FBCB2 systems. The position reports provided by these systems are disseminated across the architecture providing what is called Situational Awareness (SA).

The general concept behind ABCS is to take a large number of disparate systems, establish a secure network as the conduit for communications, institute a standardized messaging format, and disseminate C2 messages and SA across the architecture as a facilitator of combat effectiveness. The value of such a system is intuitively powerful, and analogies are easily drawn to the Homeland Security problem domain, thereby suggesting that lessons learned will have direct applicability.

3.1. Lesson Learned – Have Adequate Initialization Data Available to Support Security Requirements

Any SoS solution that provides the C2 backbone for mission critical systems must be secure. Once a system becomes compromised, whether perceived or real, the system loses effectiveness. For a system to be viable, the commander must have absolute trust in the information being provided. The maintenance of data integrity is multifaceted, and one facet is the security of the data being transmitted. Initialization data plays an important role in this security.

Every reasonable measure must be taken to secure the information that is being transmitted, and to prevent the infusion of misinformation. Security must be looked at comprehensively, and there are many aspects to consider. It is an intrinsically bad idea to have only one line of defense. The first line of defense is usually physical security, which itself, contains multiple layers. However, with wireless networks, physical security is difficult to maintain and can be easily breached. The next line of defense is encrypted authentication and authorization, followed by information encryption. An integral part of the latter lines of defense is to know who can, and cannot be on the network. A network is most secure when it is predetermined which resources can, and cannot connect to the network, and from where that resource can connect. Anything outside of that fixed⁷ configuration is denied access.

⁷ This does not preclude a dynamic operational environment, but that discussion is beyond the scope of this paper.

The reason for highlighting this last point is that the connection between strong security, and the availability of network configuration, along with resource data is not intuitive to some. To illustrate this concept, let's take three separate network environments and discuss briefly the security provided, and the data required to implement each. The three networks are at a coffee shop, a home network, and an office environment.

For the first example, in a coffee shop the security is generally very loose. Anyone can come in and connect to the network. However, the user probably cannot access the Internet unless that service is provided free. Usually the user needs an account from a carrier such as T-Mobile, or Sprint. However, even without an Internet account, the user can connect to the local network established at the coffee shop and could, for example, begin snooping unencrypted traffic. A sophisticated user could easily take advantage of a less sophisticated user in these environments.

In a typical home network configuration, one might want to set up the network where family can connect from anywhere in the house. Ideally there is an extra level of security that prevents just anyone from connecting to the local network. This can be accomplished by requiring an encrypted password to connect to the local wireless network, and can be further restricted by filtering by Media Access Control (MAC) address. It might also be desirable to restrict which Internet sites can be accessed. This can be accomplished by setting up parental controls. Typically, this provides a relatively secure⁸ environment from intruders and protects access to the Internet service.

Lastly, the office environment will be discussed. Depending on the level of security required, wireless networks would probably not be allowed. There would likely be additional firewall restrictions and filters that not only help control which computers can connect to the network, but also from where they are allowed to connect. The user can be restricted from connecting to the Local Area Network (LAN), or the Internet, or either one. All of these restriction options presuppose some knowledge of the user or the user's computer, and the network.

There is any number of configurations possible for each of the above scenarios, just as there are for ABCS architectures, and for incident response systems for Homeland Security. The ability to apply various levels of security is dependent on the availability of predetermined data, balanced with the desired level of security.

As the level of security increases, so does the amount of data needed to enforce that security. The more security

⁸ Obviously, there is more to do in the way of virus protection and firewalls, but the focus here is controlling access by predetermined configuration.

desired, the more data needed, and the more preplanning required. A valuable lesson to take away here is that a secure system requires initialization data.

3.1.1. Why would an Emergency Response System Be Concerned About Security?

This has almost become cliché since 9/11, but the world has changed since 2001 and so must our tactics in dealing with terrorists. It is assumed that any incident response system developed today must be useable in response to a terrorist attack. Furthermore, it must not be assumed that first responders will always know when they are responding to a terrorist attack. It is not uncommon for the first responders to be the real target of a terrorist attack. Therefore, if it can be thought of, the terrorists can think of it, and it is our responsibility to mitigate that risk.

Several examples come to mind of ways that measures intended for protection could be used as weapons of terror. In California, there was a proposed bill that would require all trucks hauling hazardous materials to have a device installed that would allow law enforcement to take control of the vehicle. However, it was successfully argued that it would not take too experienced of a terrorist to override the device and use it for the very purpose it was intended to prevent. The bill was rejected⁹.

There are numerous other cases where defenses can be used as weapons against the defenders, and automation systems are no exception. A system intended to enable reliable information exchange could be used to inject misinformation, intercept messages, block message traffic by overloading the system, and, as a minimum, cause confusion. Therefore, it must be assumed that incident response systems should be designed with security in mind.

3.2. Lessons Learned – Select an Information Exchange Data Model and Mandate its Use

Another facet of data integrity is the reliable exchange of information that in turn is dependent on an array of design considerations, standards, and technologies. However, the most important lesson here is to define an information exchange data model (IEDM), and mandate its use across the architecture.

There is a marked difference between enterprise level and application level data engineering. Enterprise level data engineering is extremely dependent on the institution of information exchange mechanisms. One of

⁹ "Throwing money at technology", By Robert Lemos and Mike Yamamoto, Staff Writers, CNET News.com, October 18, 2004, 4:00 AM PDT

the primary enablers of reliable information exchange in an enterprise environment is an agreed upon IEDM.

This problem has received some attention from HLS, and there are several candidates being considered as an Emergency Data Exchange Language (EXDL). However, history in the ABCS community has shown that this topic always turns into a politically charged debate over the values of several models. Invariably, no model gets fully accepted and “reliable information exchange” suffers.

State governments need not wait on the federal government to come up with an IEDM. The beauty of these models is that once a standard is adopted, it is a relatively simple matter to provide translation to other similar models. For instance, the state of Texas could have one model, and the state of Oklahoma another. Writing a translator between the two is not a significant problem. However, it does become a problem if a translator must be written for every county in the state of Texas and Oklahoma. Therefore, at the state level the governments must institute an approved IEDM schema for information exchange, and all approved systems must be required to use it.

This cannot be reiterated enough, the recommendation is to select a model and make it mandatory. The model can be extended as needed, and translators written much cheaper than software applications can be rewritten to adapt to the latest model. Most importantly though, reliable information exchange, and to some extent interoperability, will be significantly impaired until the information exchange mechanisms are clearly defined and enforced.

3.3. Lesson Learned – Understand the Importance of Data Management and Address the Problem Early in the Design

Before going into the lesson learned, and the Army’s solution, it would be more useful to provide a few examples and then explain how the proposed solution will solve the problem.

Using an ABCS scenario for an example, a unit commander has set up a perimeter defense. The unit has been task organized with some coalition forces. Unfortunately, the coalition vehicles are not equipped with FBCB2 systems; therefore their location does not appear on the SA maps. This creates a problem since they are widely dispersed, and the commander is concerned about the potential for fratricide.

The commander directs his staff to add the coalition forces to the system so their locations show up on the map. The staff follows orders, however, they have no

way of determining which names and identifiers¹⁰ are available other than looking in their local database. Fortunately, these are more savvy end users than others, and they successfully get the names and identifiers into the database. The staff then posts a position report for the coalition forces, causing them to be depicted on local SA maps.

Now other systems depict the icons popping up on their SA, but the coalition forces are showing up as unknown because they do not have these units in their local database. An astute field engineer recognizes the problem and researches to determine the unit’s name, and adds them to their local database. Unfortunately, this engineer uses a slightly different spelling and abbreviations. This same thing occurs on several systems.

Another astute field engineer is told to create some additional system roles and to put in some ad hoc units for a future exercise. However, the system this engineer is working on does not have the previously added coalition vehicles and units. The engineer finds what appear to be available URNs in the local database, and creates the necessary records.

The outcome from this scenario is simply chaos, and it happens all of the time with deployed systems. Now the architecture contains duplicate and unknown records throughout. When reports are sent out, which unit or vehicle location is being modified? Sometimes the consequences can be broken communications, and other times the consequence is erroneous and confusing SA. Fortunately, there have been no documented cases of fratricide from this type of problem, but this is likely just a matter of time.

Another data management scenario occurs when a unit is moving in to replace the outgoing unit, or when reinforcing a unit. The incoming system databases are not synchronized with the gaining unit’s databases and there are numerous conflicts. The potential exists for the previously created coalition force to have the same URN assignment as an incoming unit. From the example above, assume that one of the coalition vehicles was assigned a URN of 19001, and the 2 Infantry Division that is coming in has a Brigade commander vehicle with a URN of 19001. As position reports get passed around, an update is received for something with a URN of 19001. Which vehicle position gets updated on each system map?

This is just scratching the surface of the problem. What about icons and symbols being incorrect? For instance, in one database a unit symbol shows it to be an Infantry Company, but in reality it is a Field Artillery unit. The command could think there is an Infantry

¹⁰ Primary keys used in ABCS systems are called Unit Reference Numbers – URN – but are nothing more than a primary key identifier for purposes of this discussion.

Company on its flank only to find out it is a Field Artillery unit. The risk of misinformation as a result of conflicting data between databases is immense.

In a HLS scenario, the impact would be significant if the incident commander was uncertain as to whether there was an ambulance, a Red Cross station, a National Guard unit, a police car, a fire truck, or a Decontamination Team at the scene of a HAZMAT spill.

The risk mitigations for this problem are to have an initialization system that is responsible for creating the incident response data, and to have that system be the only system authorized to add or remove resources. In actuality, the system performs more functions than just initialization, and is more accurately described as a data management system that must remain synchronized with other data management systems across the whole organization, not just within the standalone architecture. It is critical that the process of data management be addressed early in the design phases, and strict governance be enforced to ensure the integrity of data.

As lessons go, this is one of the most important. Not only will proper data management correct and prevent numerous operational problems that currently plague many large-scale enterprise applications, but it also serves as the foundation for reliable information exchange, as well as the enabler of evolutionary systems.

3.4. Lesson Learned – Mandate the Use of Authoritative Data Sources (ADS)

As described in the previous section, one cause of the data management problem is that authority for data is not clearly defined. To expand on that idea, when the commander had subordinates add coalition data, the problem was that there was not an authoritative data source for coalition data. There was no system in the architecture with the requirement for managing operational data, such as adding new unit and vehicle data to the architecture.

There are some key distinctions that must be understood. There is operational data, there is static (or reference data), and there is initialization data.

Operational data provides information that is dynamic in nature, such as the status for a unit, or the location. Static data can also be called reference data. It does not change. For example, a unit always exists and most of its associated attributes never change. The unit name for 2 Infantry Division (2 ID) never changes¹¹, nor does the symbol that would represent it, or its URN identifier. This is all reference, or static data.

Initialization data is a hybrid between static and operational data. Any data that is needed to allow 2 ID to

¹¹ There is an action called “re-flagging” where unit names change in field, but that discussion is out of scope for this paper.

join the architecture, such as IP addresses, network configuration/system distribution, and the organization of 2ID within the theatre of operations are all examples of initialization data.

However, in a dynamic architecture, the organization of 2ID can change, and the distribution of systems can change, along with the associated IP addresses, but the static data does not change. Examples as described above would be the unit name, symbol code, and primary key identifiers.

If each of these data types is understood, it becomes easier to understand where the authority for that data is derived. For example, nobody should go in and change the location of a vehicle that is being reported by a Global Positioning System (GPS) device. It is assumed that the GPS is the ADS for that vehicle’s location. It would also be assumed that a unit’s status could only be updated by the unit commander, or delegated representative. Therefore, it is critical that each type of data have an established ownership and assigned authority for additions and modifications.

3.5. Lesson Learned – Normalization Applies to More than Just Data

Normalization is typically thought of as a database term, and in this context it is a precise method based on the mathematical constructs of relations designed to avoid certain data anomalies. This concept can be very complicated once you get past what is referred to as 3rd Normal Form. However, to make the point the following simple example should suffice. A company database should have only one record for each employee, and that employee should only be capable of being associated with one date of birth. Obviously, a person can have only one date of birth.

The concept of normalization extends beyond data though, and can be applied to rules and processes as well. For instance, a rule intended to restrict the number of connections to a given router should only exist once. Otherwise, there is the risk that the two rules will be implemented differently, with different results. As with rules, a process for task organizing available resources should exist in only one place. Aside from the possibility of two rules, or processes being implemented differently, there is an added risk that modifications to one will not be replicated to the other.

This might be a statement of the obvious, but it is emphasized as a lesson learned because this is a major problem in the context of a SoS architecture. The design of any enterprise system must take the concept of normalization to a new level and this simple concept is often ignored.

4. Conclusion

Proper design of the architecture is essential to the success of any enterprise level application. In developing this enterprise services based architecture the following recommendations are offered from lessons learned:

1. Have adequate initialization data available to support security requirements.
2. Select an information exchange data model and mandate its use.

Biography: Donald W. Jones served as the Senior Vice President for Chapter Services, and Vice President, Disaster Services of the American Red Cross from September 1991 until March 1, 2001.

As the Senior Vice President for Chapter Services, Mr. Jones exercised overall corporate management responsibility for the operation of chapters and the delivery of Red Cross services in the chapter lines of service, i.e., Disaster Services, Armed Forces Emergency Services, and Chapter Operations. As the Vice President, Disaster Services, he ensured expeditious and effective relief was provided to victims of disasters nationwide, and to the people of the U.S. Virgin Islands and South Pacific Island Territories, as set forth by Congressional Charter.

Most recently, Mr. Jones served as a team member evaluating the WTC terrorist attack for the Department of Homeland Security, Office of Domestic Preparedness. Prior to this assignment, Mr. Jones served as the special assistant to the President of Georgetown University, Washington, D. C. for Emergency Preparedness.

Born in 1935 in Hudgins, Kentucky, Mr. Jones holds a Bachelor of Science degree from St. Benedict's College and a Masters degree in Business Management from Central Michigan University. He was commissioned a second lieutenant in Field Artillery upon graduation

3. Understand the importance of data Management and address the problem early in the design phase.
4. Mandate the use of authoritative data sources (ADS).
5. Normalization applies to more than just databases, and the concept must be extended to rules and process management as well.

from Officer Candidate School at Fort Sill, Oklahoma in 1958. His other military education includes completion of the Field Artillery Basic and Advanced Officer Courses, the US Army Command and General Staff College, and the Army War College.

Over his military career, Mr. Jones held a wide variety of troop commanding positions as well as staff assignments. Prior to joining the Red Cross, Mr. Jones was the Deputy Assistant Secretary of Defense for Military Manpower and Personnel Policy, Office of the Secretary of Defense. His other key appointments were as the Assistant Deputy Chief of Staff for Personnel, Headquarters, United States Army, and as the Commanding General, Military Personnel Command. He was also Assistant Division Commander of the 1st Cavalry Division at Fort Hood, Texas, from 1983-85. Mr. Jones served in the United States Army for over 35 years and retired in August 1991 with the permanent rank of Lieutenant General.

Mr. Jones' awards and decorations include the Defense Distinguished Service Medal, the Army Distinguished Service Medal, the Legion of Merit, the Bronze Star, and the Meritorious Service Medal. He is married to the former Betty Karnes, and they have two children, Lori and Donald.