Appendix G

The Eller HADRIM Framework: "Humanitarian Assistance and Disaster Recovery Information Management Model"

Authors

A. Riley Eller
Eric Rasmussen, MD, MDM, FACP, Managing Director, Infinitum Humanitarian Systems

Published in Burns, R. and Shanley, L.A. 2013. Connecting Grassroots to Government for Disaster Management: Workshop Summary. Washington, DC: Commons Lab of the Woodrow Wilson International Center for Scholars

14 SEPTEMBER 2012

Abstract

This paper presents a model for secure information management in complex, multi-agency humanitarian assistance and disaster relief (HADR) missions. To the greatest extent possible, the operations that comprise this plan are transparent to both the response teams in the field and the affected community they are supporting. This puts the responsibility for an agency's information technology management processes dominantly within the home office where there is time to act with deliberation.

Motivation

The success of HADR missions is proportional to the quality of information available to relief coordinators at any given moment. To the extent that information technology can achieve high levels of quality, with a rated Force Effectiveness Multiplier (FEM) greater than 1.0, it should be deployed with all due haste. Unfortunately there are many more ways for data feeds, support tools, and reporting requirements to unintentionally decrease effectiveness with FEM less than 1.0. Therefore, coordinators must demand a quality process to ensure that every information element proposed — hardware, software, human, and process — can be shown to increase effectiveness reliably. The

consequence of a careful information management program is that the operation can consistently eliminate obstacles to success through this agile security process.

Objective

Based upon the need to utilize information technology to improve force effectiveness, we conclude that the goal of this model is to

"Control the impact of information on force effectiveness."

Methodology

Taking a page from formal control theory, we must recognize that the variable under control (FEM) cannot be changed more quickly than the manager can detect the effects of decisions made. Therefore, the process must operate on a sufficiently long cycle that new policies are not judged prematurely. This leads to a three stage, iterative model:

- Setting policy,
- Executing the plan, and
- Measuring the results as evidence for the next iteration

In addition to executing the plan in situ, we need to recognize that preparation is equally relevant. Specifically, we must prepare through the following activities:a

- Material caching,
- Budgeting,
- Developing relationships, and
- Surveying the physical and digital landscape to understand the changing world.

Finally, because plans are only as good as they are flexible, it is important to understand that the mission command staff are always the final arbiter of correct actions. This may be the case even in direct contravention of any plan element. While this surely poses a risk to the implementation of information security, it is a frank assessment of the nature of the mission and must not be "toughened" during process review. If policy authors would mandate specific behavior, they must persuade the mission staff by clear presentation of historical evidence, it is incumbent upon the writer to educate those staffers with evidence and reasoning.

Security domains

In order to implement access controls, we must first define certain domains between which boundaries will be constructed and secured. These security domains may be virtual, as in the contents of a database, or they may be physical, like a data center. Generally, virtual domains have an inherent reliance upon the physical security that

prevents access to the machines which host the sensitive information. Since the purpose of this framework is to maximally leverage information, the foundation of all security described herein should be seen as a pairing of restricted access to information and extremely restricted access to physical hosting areas.

1. Affected area

During a HADR mission, the most general security area is the affected terrain. This framework assumes that no access control to this area can be achieved. While the majority people in this domain are probably in need of assistance during the recovery effort, we also assume that some groups in the area may have malicious intent toward the mission staff and any volunteers who would assist.

2. Mission encampment

For the safety of all concerned, we must restrict access to personnel, supplies, and the command center. With an established perimeter and appropriate entrance screening, we can operate in the affected community and yet control interaction with adversarial parties. Securing this domain should be an ongoing effort, beginning immediately upon deployment and evolving along with the situation. To secure this domain, consider the following list of concerns and mitigations:

a. Perimeter incursion - fencing

b. Reconnaissance from without — opaque fencing

c. Crossing the fence from within — monitoring device with motion detection

d. Volunteer entry — photo identification only; passwords too hard to remember, biometrics irrefutable

e. Access probing (malicious volunteers, identifying "collaborators" who pass within) — offer a duress "button" and greatly increase caution when it is pressed.

f. Vehicular overrun — in cases where highly adversarial populations may use car bombs or other incendiary devices, follow the "green zone" model of concrete obstacles to deny vehicle access.

g. Wireless snooping - GSM, Wi-Fi, and other wireless data connections must be deployed in a secure fashion to prevent access from outside the encampment.

3. Mission operation center

Mission planning information must be secured from all non-essential personnel as it may be lethal in the hands of opposing forces. Thus, deep inside the mission encampment is the operations domain. It should be in a position that would make it as difficult as possible to reach from outside the encampment. To secure this domain, consider the following techniques:

a. Access control - Photo ID badges

b. Extremely sensitive situations - Daily password rotation, to be briefed each morning to the minimum feasible group

c. Weapon, medicine, or other highly valuable stores - 24 hour guard, potentially armed; or, secure access technology like a “man trap”.

4. Organizational headquarters

The “home office” of the leading operation team. This is usually distant from the affected area, and connected via “umbilical” data links such as shortwave radio or satellite uplink. The headquarters, at least its data center, must be at least as secure as the mission operation center. Otherwise, the intelligent adversary will simply invade the mission from a great distance. The means to secure a daily use facility is outside the scope of this document, but a majority of all defensive spending should focus on this most durable domain. Especially sensitive information regarding volunteers in affected areas MUST NEVER be stored outside this domain. Access MUST be rigorously controlled.

5. Human Information Database

The “crown jewels” of any organization are its people. Securing personally identifiable information (PII) about the staff, volunteers, staff, and affected individuals is the most important role for HADRIM. Without trustworthy protection of the people, every other goal of the HADR mission is in jeopardy. The canonical store should follow rules equivalent to the best commercial offerings; as an example, consider the Amazon One-Click system where the payment system can only be controlled from a web browser but card information cannot be retrieved. This is probably not achievable with open source tools and best effort planning, but instead requires very diligent implementation by an experienced security engineer or architect.

Roles

This model assumes that involved parties are already busy with their work. To implement these recommendations, then, requires HADR teams to increase their ranks by one member. The new Integration Engineer role is complex and nuanced, and should be seen as a technical leadership career. These technical managers need to understand subjects as diverse as the Incident Command System, UN relief agency charters, international response team mandates and resources (e.g. the Icelandic Urban Search and Rescue Team, the Israeli Eye Injury Management Teams, the US Disaster Mortuary Assistance Teams), local transportation, communications, and data capabilities, network engineering, recognized inter-agency rivalries, recurrent response team personnel, collaboration and mediation skills, physical self-reliance, personal and data security protocols, media crucible techniques, and agile software development for field conditions. Agencies need to develop individuals with this level of training.

With the addition of the HADR Integration Engineer, much of what follows becomes feasible.

Stage 1: Preparation

Preparing for situational information management is broken into three distinct activities with associated goals:

Stage 1:

Activity 1.a: Software design

Developing the ability to visualize data as needed by the deployed staff. Any deployed software must be maintained by a durable entity such as a commercial, governmental, or non-profit agency. No software can be deployed that violates this rule without unacceptable risk.

Activity 1.b: Data surveillance

As there are many durable sources of information on which HADR missions rely, especially weather and geospatial information services, it is valuable to fully integrate these sources with the software developed in Activity 1.a above. However, many other data sources are more dynamic than agency process can manage. For that reason, it is crucial that the software be configurable by technicians in the field to use ad hoc data sources as they are discovered. And since any method of data access, like a given NOAA web service, may fail, it is important that field-selected data sources can be configured flexibly and with minimal technical skill.

Activity 1.c: Relationship management

Trust is an important characteristic of successful missions; effectiveness drops when there is mistrust between the people involved at any level and on any topic. Managing relationships with the many people involved in potential future missions can protect the software, information, and people in the affected area. Education and role-playing in collaboration and mediation, formal agreements, and Customer Relationship Management (CRM) software can each be used to assist with this crucial and under-supported activity.

Goal 1.c.11: Engage software developers

As all software used in this model must be managed by a trustworthy entity, it is crucial to connect volunteer software developers with those entities early and with clear development standards for the software engineers involved on both sides. A reasonable standard would be reaching out to every relevant developer at least once each year. Encourage development organizations to adopt a security maturity model (such as the Building Security In Maturity Model) by preferring contributions made by more mature contributors if the options are otherwise equivalent.

Goal 1.c.2: Engage open data providers

Staying abreast of developments in the open data movement and digital sensor market is crucial if Activity 2 is to succeed. Challenging volunteers to test any agency's

assumptions about each data type and source is a sensible means of improving awareness and preparedness as capabilities change. Test the information catalog at least once each year. Audit each data source at least once to validate that appropriate security, redundancy, virtualization, and/or field deployability claims can be substantiated.

Goal 1.c.3: Engage global volunteers
Identifying and connecting with volunteers around the globe can create the seeds of trust by liaising between responders and the affected community. Online forums, chat rooms, video conferences, and video games can be used to increase the sense of community and engagement. Reach out to every member of a global volunteer seed network three or four times each year. Give the most active contributors additional responsibility to coordinate with others in their area for training and sandbox exercises.

Stage 2: Integration

Every response has unique features that contain context for information. One simply cannot reliably predict which data sources will be available or how each information element should be interpreted in a given mission. Therefore, most of the data sources connected to visualization software must be connected in an ad hoc fashion.

Of course, a few durable sources like weather should be pre-configured by default with highly reliable and independently maintained source feeds. To manage those data sources that appear during the response, become indispensable, and were not known before deployment, one or more Integration Engineers must be deployed with a response team to integrate the prepared tools with available data sources and feed those to other relevant teams throughout the response.

Activity 2.a: Data Reconnaissance
By continuously re-evaluating the data sources already cataloged in Activity 1.b, the Integration Engineer can develop a situation-specific information catalog current at the onset of any event. This will be the scope of the data that will be available at the onset of the mission. It should be briefed in an accessible and replicable format as “Best Available” to teams attending the first field-based Humanitarian Update brief.

Activity 2.b: Software Integration
As an Incident Commander or volunteer recognizes that a given view or function will be of use for the mission, that should be passed as a requirement to the Integration Engineer. Once the tool has been connected to the appropriate data sources, the working software can be deployed and briefed to fellow responders as a resource.

Activity 2.c: Activate Local Volunteers
Communicate the extents and goals of the mission to the community developed in Activity 1.c so that local volunteers can quickly engage the affected population and begin to develop the relevant lines of local communication. This cannot proceed until

the integration stage is relatively complete; the volunteers must be connected to the process only after it is up and running.

Goal 2.a.1: Rapid Command Activation
Upon deployment, the mission commander must produce a list of necessary catalog elements. Within 12 hours, integration of this first set of tools should be complete and handed off to the commander.

Goal 2.a.2: Timely Completion
Within 24 hours of deployment, all of the remaining tools listed in the catalog should be integrated and delivered.

Stage 3: Implementation

After preparing and integrating, the mission proceeds.

Activity 3.a: Technical Logging
Every information technology component (software and hardware both) must support high resolution activity logs for post hoc analysis. Every action performed by each user must be recorded. Every computer-to-computer interface message SHOULD also be recorded.

Activity 3.b: Event Logging
To provide context to the technical log, the human activity log must also be made available to analysts after the fact in Stage 4.

Goal 3: Failures must be captured
While it is impossible to predict how a given scenario will interfere with the best laid plans, the goal of logging is to capture the historical record in sufficient detail that most problems can be observed in the log.

Stage 4: Analysis

Discover inefficiencies and failures of the model. Recommend changes for the next iteration.

Activity 4.a: Historical reconstruction
Technicians transform the human and machine logs into a single narrative that attempts to capture the sense of the situation rather than every precise detail. Each input fact and narrative element must be connected in a manner that can later be used for forensic analysis.

Activity 4.b: Information management process review
Assemble a panel from members of the mission as well as software developers, data providers, and volunteers from the affected community. The committee's role is to provide commentary and guidance for improving the effectiveness of human, software, and data elements of future missions.

Activity 4.c: Corrective improvement
Implement the recommendations of the review committee.

Goal 4: Timely guidance for future missions
The review committee should seek to meet early, work with due urgency, and present their findings quickly. Ideally, this report will be made available to all parties within 90 days of the first de-escalation of each mission.

Key Performance Indicators

The efficacy of this process must also be evaluated and improved over time. As such, it must generate useful metrics along the following lines. Success Thresholds should be re-evaluated periodically, as a quality process will improve over time. The values presented here are mere suggestions. These indicators do not and cannot define success! Success can only be found in the health and well-being of the affected population. Instead, use measures like these to help identify process weaknesses.

Indicator 1: Software uptime

Total number of hours of proper software function for each component, divided by the duration of the mission. To be computed by comparing service start and stop events in the event log.

Success Threshold: 95%

Indicator 2: Information availability

For each data element recorded in the Activity I.b catalog, the proportion of hours that source was available to the mission. To be computed as the proportion of successful data source connection requests compared to all data source connection requests.

Success Threshold: 95%

Indicator 3: Information reliability

Proportion of data elements requested by software to the number of those elements delivered without error. To be computed as the proportion of successful data access requests to the total number of data access requests.

Success Threshold: 99%

Indicator 4: Volunteer activation time

The mean time between the first request for volunteer services and their arrival at a determined gathering location. To be calculated from the volunteer check-in log.

Success Threshold: 48 hours

Indicator 5: Decision latency

The mean time between data availability (which begins when a useful fact first arrives at a software tool through a data interface) and the first activity recorded that makes use of each datum. To be computed as the average time between commands issued and the most recent event presented in the user interface at the time of each command.

Success Threshold: 5 minutes

Indicator 6: Command activation time

The interval from first arrival on scene until completion II.a.

Success Threshold: 6 hours maximum

Indicator 7: Data-tool integration latency

Time to integrate each data source with each software tool. To be computed as the duration between the completion of 2.a and the completion of 2.b, divided by the number of data source and application interconnections.

Success Threshold: 30 minutes per connection

Indicator 8: Logging utility

The proportion of failures or defective behaviors reported that can be accurately reconstructed from the event log. Ideally, the log will permit complete, accurate forensic reconstruction of each failure. To be computed from the post hoc software development activities by polling the software developers and/or quality engineers.

Success Threshold: 95%