RESPOND-R Test Instrument

A Summer Institute 2013 Case Study

Brandon Shrewsbury, Zachary Henkel, Chang Young Kim, Robin R. Murphy
Department of Computer Science and Engineering
College Station, TX , USA
{btshrewsbury,zhenkel,kcyoung,murphy}@cse.tamu.edu

Abstract— **The RESPOND-R framework is a modular, distributed data logging framework built for the purpose of studying Emergency Informatics as it relates to human-robot performance during disaster responses. Currently deployed systems have limited data logging capabilities, with few incorporating heterogeneous robotic systems. Current systems focus on human performance and event information, but often incorporate restricted ontologies that inhibit real-time sensor logging of robotic systems. RESPOND-R provides broad logging coverage of robotic systems and human performance data, bridging the gap between the heterogeneous loggers and current emergency management systems. A Packbot 510 UGV, an AirRobot 100B sUAS, an AEOS Marcy USV, and nuclear biological chemical (NBC) sensors have been integrated into RESPOND-R. The paper describes how RESPOND-R supported data collection during a 2013 radiological forensics field exercise. The data has been used to analyze autonomous flight patterns for nuclear forensics and to isolate and identify the cause of a sensing anomaly.**

Keywords— logging, disaster, ros, emergency informatics, data to decision

I. Introduction

Cyber-Physical Systems (CPS) that include robots, wireless networks, and sensor networks are currently under evaluation for use in emergency preparedness, prevention, response, and recovery. However, a lack of comprehensive integration into larger emergency informatics systems has prevented a thorough investigation of the interactions between the devices, the physical world, and human decision-makers. The RESPOND-R framework addresses this problem by defining an open standard for the logging of heterogeneous device data and a method for storing, assembling, and reviewing captured data.

The RESPOND -R framework acquires data from CPS components in four core areas: unmanned systems, imaging, sensors, and human performance measurement devices. The majority of the data is logged automatically without human intervention, though the system is capable of accepting human input as part of the logging data. This automation of logging all cyber physical system data relieves the burden of manual collection and management of event data. Following information collection, the framework aids in the digestion and context classification of logged material. The result is an integrated, interoperable research instrument suitable for comprehensive performance analysis of CPS technologies and post deployment data management.

II. Ease of Use

Pivotal to both ensuring a successful response and to providing data for later analysis, disaster data management strategies have been the subject of much research. While numerous disciplines have contributed to improving the landscape of disaster data management tools, none have examined the importance of consistently logging data from a group of heterogeneous robots deployed in the field within the context of disaster management. Two separate efforts are relevant to this work: general strategies for data management in disaster situations and data logging strategies for heterogeneous groups of robots.

The disaster data management domain grapples with a variety of difficulties, including a large number of data producers, a lack of common terminology among systems, and a variety of data formats [2]. A survey of data management solutions for disaster situations by Hristidis et al. highlights five different systems previously used by disaster responders [2]. The systems examined focus primarily on GIS capabilities, message routing, resource management, and data tracking. These systems receive human input as data, rather than machine sensor data. For example, the Sahana disaster management tool uses a component based design to support organizational efforts, inventory management, and asynchronous merging of data [1]. While these systems focus on human collected data, many of the techniques employed are transferable to data collected by robots. For example, Currion et al. describe the need to be able to use non-networked mechanisms for transferring logged data when network connectivity is unavailable [1]. Furthermore, Lezcano et al. elaborate on the difficulties that can occur when integrating data sources from different groups [3]. The lack of semantic equality between heterogeneous systems can greatly impede disaster response efforts. Current standard protocols, like the XML based TSO standard, lack support for integrating multiple languages. While the system describes syntactic templates for data integration, semantic meaning can be easily lost. This draws attention to the lack of interoperability between a heterogeneous robot data logging system and existing disaster data management solutions.

978-1-4799-0880-6/13/$31.00 ©2013 IEEE.

Both Michael et al. and Lopez et al. have explored data logging or data sharing for heterogeneous groups of robots, though not in a disaster scenario [4,5]. Additionally, both utilized a messaging protocol agreed upon a priori and programmed into each robot. While this scenario does not accurately reflect the nature of disaster response data, it does allow for a demonstration of differing communication strategies. While Lopez et al. primarily examined coordination of tasks among multiple robots; their architecture highlights the importance of modularity in design [5]. Modularity allows various components of the system to malfunction without halting the overall operation of the larger system. Data logging for a group of robots benefits directly from using a modular design with components that are independent of one another. However, the design implemented in Lopez et al. used a centralized server to route messages between robots, an approach which allows for a single point of failure. The fragile nature of communications networks during disaster responses produces a need for systems which can operate in a peer-to-peer or asynchronous nature as a central server may not always be accessible. Additionally, it is often the case that network connectivity is unavailable, yielding the requirement that responders be able to upload data to a centralized server at a later point in the operation. Michael et al. illustrated the importance of utilizing both local and centralized data logging within their experimental multi-robot system [4]. Despite being designed to function indoors with reasonable bandwidth rates, the design of the data logging system provided for both local and centralized logging of robot data. In this design, local and remote storage capabilities were used in a complementary fashion in order to utilize storage resources appropriately. This implementation ensured that logging data was not lost due to network connectivity issues. The findings from these two studies suggest that modularity and connectivity are among the most important issues in logging data from multi-robot systems. Importance should be placed on allowing the system to continue logging successfully even when connectivity issues occur or when individual components fail.

Together, prior work suggests that data logging should be performed in a distributed and asynchronous fashion. The likelihood of low bandwidth network connectivity or no network connectivity should also be considered in the overall system design. Finally, interoperability between the data logging system and other systems must also be considered as other systems may become consumers of the data produced..

III. RESPOND-R FRAMEWORK

In an effort to address a lack of dependable network infrastructure, which is typical in disaster scenarios, the Respond-R framework was designed with a primary focus on decentralization with an option for centralized routing. This approach included creating a model for distributed logging, creating a sensor storage model, and determining a naming convention for standardization.

A. Decentralized Logging

The decentralized system was designed to require each CPS to communicate with an independent logger that stores CPS states locally. The logging component must be able to store as much information as is available through CPS interfaces. In some instances little or no data is collectable by the CPS due to manufacturer design. In these instances visual data or manual annotations must be recorded with additional CPSs used to augment the lack of necessary context data. Gathered data is stored locally while a real-time propagation component can relay information when network infrastructure exists.

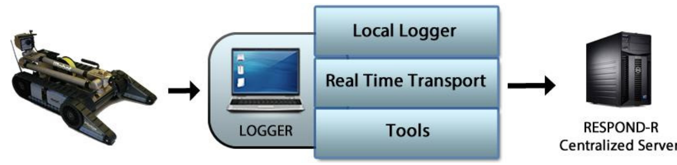

Fig. 1. Overview of a Logging Component

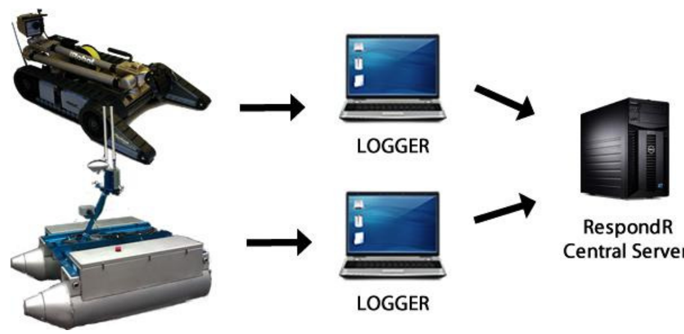

Fig. 2. Sample Logging System

B. CPS State Model

For each CPS, a minimum set of fields (location, time, and sensor readings) are required to evaluate performance, though generally this minimum is not enough to properly evaluate the data-to -decision process. Environmental factors impacting the human operator must also be captured. This includes orientation, video, audio, signal strength, and user input. Video feeds from the CPS should be logged for environmental factors affecting the CPS and video data of the user interacting with the control interface, including both exterior views and screen views should be captured. Failure analysis requires the logging of wireless strength, IMU, altitude, and battery data when available. Figure 3.2 depicts typical fields captured from an unmanned system. Much of this data cannot be captured from CPS hardware alone. The logger's internal hardware, such as an integrated camera or microphone, can be used as an extension of the CPS sensor set. A combination of CPSs can aid in the environmental gap. From past experience, wearable video cameras on the operator have proven to be the most effective way to capture the environmental information missed by typical CPSs.

Due to personal experience and device trust from frequent use, the "bring your own device" approach is employed by responders and creates a taxing development issue within the logging domain. Each CPS device must be individually integrated into the system for proper coverage. This cannot be done prior to deployment due to device variability. During real world responses, onsite integration of new systems is infeasible. Thus, the development cycle must continuously integrate new devices, calling for standard data interfaces from commercial products. Until this is done, an iterative approach is required to incorporate new equipment and redefine methodologies. Continuous refinement of the logger for future deployments as well as the tailoring of the logging approach for verifying specific claims allows for a more robust and usable system. Although the authors realize this approach is not novel it should be noted that the infancy of this field requires maintenance of the logging systems to thoroughly validate claims.

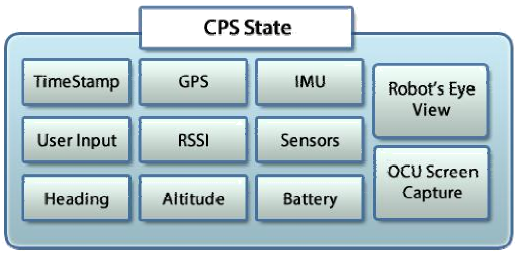

Fig. 3. Typical fields logged from a teleoperated wireless unmanned system

C. *Naming Convention*

Focus is placed on tagging and indexing the digital storage format – the file – as the principle quanta of data for curation and as a practical starting point for creating a consistent, replicable organization of a data archive. The proposed file naming convention allows for indexing and queries using native search. This approach limits the overhead of external cataloging but constrains the search methods and relies heavily on consistent indexing across operating systems. A set of five surface attributes, which are easy to determine without deep understanding of the content, are the basis for the proposed native organization and access methodology:

Data type: primary or secondary - File names will begin with a "primary" or "secondary" type attribute defining the Type of data. Primary types originate from an unmanned system, human, or sensor such as UAV while secondary types represent context data, such as an external picture of a Person performing an action.

Source: provenance (from a specific device, human observer, etc) - The source is divided between super-type and subtype for indexing purposes. Super types include general classification such as UGV or Responder. Subtype includes a unique identification of the agent such as Packbot or Aeos-Marci. Some data will have no subtype.

Semantic content: places, agents, devices, activities - The semantic content describes the content of the data. A generic category is supplied first such as place, activity, or image with a brief description of the data such as image-overhead-hospital.

Date and time interval: A timestamp and interval is supplied in the format YYYY -MM -DD-hh-mm-ss-iiii. Although timestamps exist with files, files are transferred via multiple forms and may lose metadata through the process. This approach ensures time of creation is not lost. An interval is included in minutes if meaningful

Location: Geotagging is done using the United States National Grid system (USNG). USNG is defined as the common coordinate system for disaster response. It is also advantageous for use in naming as it uses a more compact than the latitude longitude representation

Additional constraints ensure further data consistency:

1) *Lowercase*
2) *Use underscores as element delimiter.*
3) *Use hyphens to delimit words within an element*

The categorization of files by surface attributes has several advantages. It provides consistency. The categories are general purpose and apply to decision-making domains in general, not just disaster response. The categories lead to tagging by filename conventions expressing the code for each attribute value. These tags may be automatically generated as tools are developed to support data capture. The disadvantage of the categories is their coarse granularity. However, it is expected that more advanced tools for snippeting and synchronization will evolve once the community has the opportunity to work with consistent datasets and determine what capabilities are most warranted.

People are often not able to or not willing to engage in data classification activities [9]. Tools that automatically capture or provide support for the capture of said surface attributes could prove vital to the building of valuable data collections for research. When it is not possible to automate the capture of the data and its features, tools will be employed that opportunistically support the incremental recording of the attributes during normal user activity [10] and improve consistency of attribute/metadata use [11].

The example listed below depicts the Packbot 510 primary camera taking picture of a derailed train car during a field exercise. Notice the interval has been omitted.

primary_ugv-packbot_image-boxcar-3_2012-05-16-10-41-46_14rqu5401085519.jpg

IV. RESPOND-R LOGGER IMPLEMENTATION

RESPOND-R Test Instrument consists of a collection of Robot Operating System (ROS) nodes. The ROS architecture was chosen for its open source availability, existing data storage, and replay libraries. The nodes are divided into three sections: data collectors, data management, and data exporters. The data collectors are comprised of independently tailored nodes for each CPS system strictly developed to parse and publish CPS states. Data management tools aid in the merging and splitting of data among various storage files. The export tools provide an easy avenue for post event analysis using systems not native to the ROS environment.

A. *Data Collectors*

Each decentralized CPS data collector retrieves data through a CPS interface and publishes states to the ROS blackboard. A CPS system may have multiple data streams and require multiple logging nodes. Each logging node contains an interface to retrieve specific CPS data, a parser for data digestion, and a publisher to send the data to the ROS blackboard. Once on the blackboard the rosbag recorder stores the information.

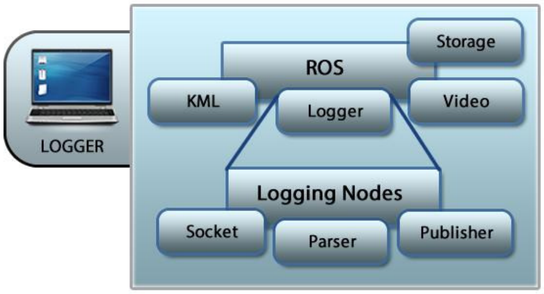

Fig. 4. Logger Architecture

B. Storage Management

Bag management reduces the time needed to sort through event data. Both merging and splitting of bags is done using the third party tool bagedit [12]. Once the event is complete logs are joined using bag merge to allow complete scenario replay. Due to size and re-indexing, incremental bag storage is recommended with splits between a consistent time.

C. Export Tools

Post event analysis is outside the scope of this paper though tools are included to facilitate the export to common formats for external viewing. GIS is a common medium for sensor data and CPS movement. An Incremental KML creation Node is included in the implementation. GPS, time stamps, altitude, heading, and sensor reading messages are exported to a KML file at a specified interval in real time. From field experience, it is common for geo -location data to be invalid though discarding the entry is not advantageous. Thus, both valid and invalid data is stored within a KML file for user interpretation.

A video stream export tool is also included in the implementation. Incoming CPS video streams are stored in the ROS format but extraction of video data from the bags in not a trivial task. Environmental factors such as network dropout and video switching will change the number of frames and skew the frame rate over time affecting the outcome. The video tool encodes the video in real time as the stream is published to the blackboard.

V. CASE STUDY

The 2013 Summer Institute provided the most comprehensive use of RESPOND-R to date and is described in this section. Early versions of the RESPOND-R system have been used to capture single robot or human -single robot interaction at the IRS-CRASAR Tohoku tsunami response [7], basic research in human-robot interaction [6], and as a complete system in the 2011 and 2012 multi-robot Summer Institutes. In the 2012 Summer Institute, RESPOND-R collected data on the data-to-decision process between a small UAS and incident command [14] and documented the emergent cooperation styles between a UAS and UGV [12].

The Summer Institute was held at Disaster City® in College Station, Texas. RESPOND-R was used for the radiological exercise conducted on June 6, 2013, which simulated the collapse of a hospital with a large nuclear medicine wing from an earthquake. The RESPOND-R test instrument interfaced with an AirRobot 100B sUAS carrying a video payload, one Packbot 510 UGV equipped with 3 onboard cameras, a Geiger counter, Biological Agent sensor, 4 gas detectors, and a AEOS Marcy USV equipped with 2 on board cameras. The USV was not fielded due to hardware difficulties.

Fig. 5. Packbot 510(Left), AirRobot 100B(Center); AEOS Marcy (Right)

A. Collected Data

Rosbags, digital images, videos, scanned documents, and transcribed notes were stored using the RESPOND-R test instrument, during Summer Institute 2013. Table 1 details of the total division of the 201 GB collected during the exercise. There were four types of data collected during the Summer Institute, these are listed below. The frequency of the robot data collected depended on the device; Packbot data was recorded every 250 milliseconds, while handwritten notes were uploaded manually at the end of the event.

1) Context: The documents related to the overall operating conditions and what was going on such as a plan of action for the day, maps in hardcopy, or videos of the site.

2) Video and still images: Video and still images captured from robots or human-robot interaction data-capture cameras.

3) Measurements: Data from all of the robots' sensors and network devices including sensor calibration, battery power, network strength, GPS, compass, accelerometer, gyro, temperature, humidity, pressure, Radiac, MultiRAE, etc.

4) Activity Logs: These were both computer generated logs of keystrokes, screen capture, etc. and handwritten ethnographic notes.

TABLE I. SIZE AND QUANTITY OF FILES COLLECTED

Type	*Number Of Files*	*Size*
Rosbags	5	181 GB
Images	714	1.91 GB
Video	30	16.4 GB
Scans	4	1 KB
Txt Files	1	0.5 MB
KML Files	3	2 MB

Datasets were stored in 13 formats for video, imagery, sensor measurements, observational data, and hard copy data. Video files were stored using the ROS uncompressed image standard while external video recorders used five file formats MPG, MP4, MOV, WMA, and AVI. Image files were stored in

JPEG, TIF, PNG and GIF, while sensor measurements were captured in TEXT/ASCII format within ROS messages. Hardcopy materials such as handouts from agencies including the plan of action for the day, maps, etc., were scanned to PDF, transcribed to ASCII, or photographed.

B. Robots

The iRobot Packbot 510 logger collected information via the Packbot developer package, the Operator Control Unit (OCU), and a video streams of the OCU screen using VLC server. Camera, chassis, and arm data was gathered directly from the robot using the iRobot Packbot Developer API. HAZMAT sensor data such as the Canberra Geiger Counter readings, multiRae gas detector readings, and the Smiths Detection Chemical warfare detector readings were not available through the Developer Package, but the data was collected through modifications within the OCU software. Both the HAZMAT data and the user input were transported through sockets to the CPS Logger.

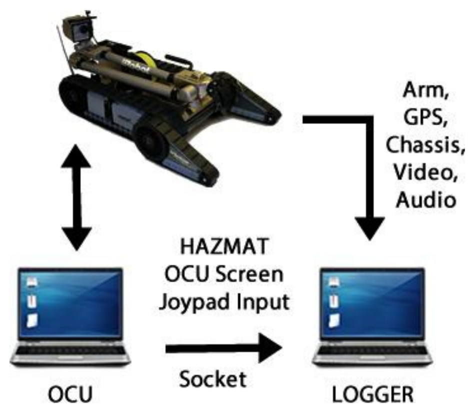

Fig. 6. Flow of information from the Packbot to the logger

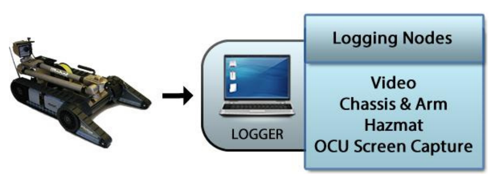

Fig. 7. Packbot Logging Nodes

The Packbot performed one continuous run during the June 6th exercise with 2 logged segments. The first segment collected 91.4GB of data during the initial 1 hour and 48 minutes of the exercise while the second segment collected 45.9 GB covering the final 1 hour and 9 minutes of the operation.

The AirRobot 100B's data was stored using an existing logging interface supplied by the manufacturer. The commercial logging component collects information via USB to serial converter and stores data in XML format. The AirRobot 100B also exports video via a virtual webcam as well as USB flash drive from the proprietary base station. An offline tool was used to combine and export video and XML data to ROS messages for storage by rosbag. The use of the offline logger proposes a particular challenge as the XML logger and video streams were created by independent systems and may not match in length or start time. This would cause mismatching of visual and non- visual data during analysis. Metadata was used to match file start times, but inconsistent metadata from non-synched system times and incomplete user logs require manual estimation of start time offsets.

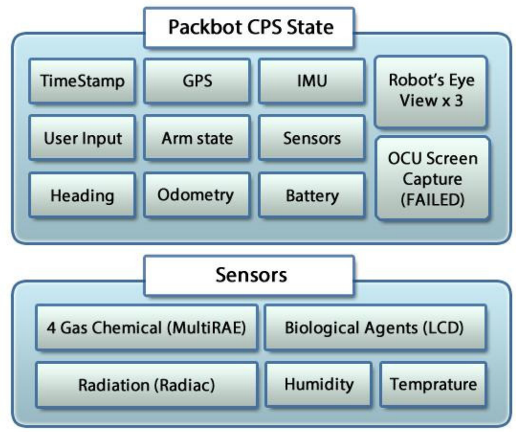

Fig. 8. States collected from the Packbot 510

The AirRobot had two flights. The first flight was an initial damage assessment with a 10-minute flight time, while the second flight operation was a radiological source forensics mission with a 14-minute flight operation. The initial flight collected 14 GB and the second collected 22GB of data.

C. Results from Preliminary Analysis

The data gathered by RESPOND-R has been used in the analysis of anomalous UGV behavior and in documenting UAV flight demands.

RESPOND-R was used to isolate and identify the unexpected impact of radiological sensor placement [14]. During the exercise, the HRI data showed that the Packbot operator commented that he was able to find the Cesium source only when driving the robot backwards. He repeated this belief during the outbrief. An analysis of the collected video, radiological, heading, and odometry data from RESPOND-R confirmed that this was the case: the radiological sensor was mounted such that the front of the robot and the manipulator arm hardware absorbed radiation, while the rear of the robot was an open cargo space. Without the frequency and cross-cutting data collection, it would have been impossible to document what happened and the unexpected behavior would have been written off as human error.

The second finding was the unexpected flight demands for sUAS. The flight paths of the UAS were captured and identified two challenges for near Earth flight [8]. First, the most productive flight paths for nuclear forensics were at altitudes between 0.3 and 30m, with the sUAS hovering at 1.5m or less from a structure. Second, the AirRobot data showed that a decrease in GPS satellite coverage reduced the ability of the flight stability software to hover in place.

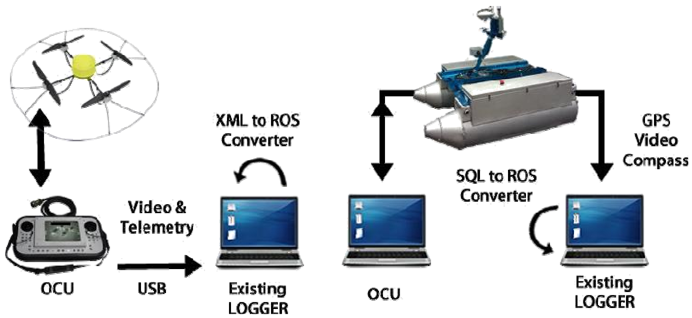

Fig. 9. Logger Configurations, AirRobot (LEFT) & USV(Right)

VI. CONCLUSIONS AND FUTURE WORK

The RESPOND-R test instrument provides an extensible, open source framework for data logging of diverse and heterogeneous robot data sets. The outlined naming convention allows for standardization and queries using native search tools, while the review of the RESPOND-R framework's role during Summer Institute 2013 has illustrated the diversity and complexity of complete data collection. The use of collected data from the framework has contributed to new findings in the disaster domain, though RESPOND-R is applicable to any human-machine system.

It should be noted that having a comprehensive logger does not guarantee complete data is available. In the case study, the RESPOND-R data was incomplete because the UGV itself had no way of providing accurate location indoors and the experiment did not have an external mechanism such as a motion capture camera. As a result, the robot location at each time step was estimated manually. RESPOND-R will become even more useful as data logging standards and test methodologies emerge. Future work will add tools for visualization and mining techniques.

The RESPOND-R framework, documentation, and collected data are available through http://respond-r.cse.tamu.edu and http://sourceforge.net/projects/respondr.

ACKNOWLEDGMENT

This work was sponsored in part by NSF Grants CNS-0923203. The authors would like to thank TEEX, Texas Task Force 1, and the Texas Military Forces Civil Support Team for their assistance at Disaster City ®.

REFERENCES

[1] Currion, P., Silva, C., & Walle, B. Van de. (2007). Open source software for disaster management. Communications of the ACM, 50(3), 61–66.

[2] Hristidis, V., Chen, S.-C., Li, T., Luis, S., & Deng, Y. (2010). Survey of data management and analysis in disaster situations. Journal of Systems and Software, 83(10), 1701–1714.

[3] Lezcano, L., Santos, L., & García-Barriocanal, E. (2013). Semantic Integration of Sensor Data and Disaster Management Systems: The Emergency Archetype Approach. International Journal of Distributed Sensor Networks, 2013, 1–11.

[4] López, J., Pérez, D., & Zalama, E. (2011). A framework for building mobile single and multi-robot applications. Robotics and Autonomous Systems, 59(3-4), 151–162.

[5] Michael, N., Fink, J., & Kumar, (2008). Experimental testbed for large multirobot teams. Robotics & Automation Magazine, March, 53–61.

[6] Murphy, R. R. (2004). Human-robot interaction in rescue robotics. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 34(2), 138–153.

[7] R. Murphy, Et Al, Marine heterogeneous multirobot systems at the great Eastern Japan Tsunami recovery, Journal of Field Robotics 29 (5) (2012) 819-831.

[8] Duncan, B. Murphy, R. Autonomous Capabilities for sUAS Conducting Radiological Response: Findings from a High Fidelity Discovery Experiment. Journal of Field Robotics Special Issues on NearAutonomy. to appear.

[9] Shipman, F. M. III and C. C. Marshall, Formality considered harmful: Experiences, emerging themes, and directions on the use of formal representationsininteractive systems," Comput. Supported Coop. Work, vol. 8, no. 4, pp. 333-352, Oct. 1999.

[10] Shipman, F. M. III and R. J. McCall, \Incremental formalization with the hyper-object substrate," ACM Trans. Inf. Syst., vol. 17, no. 2, pp. 199{227, Apr. 1999.

[11] Shipman, F. M. H. Hsieh, J. M. Moore, and A. Zacchi, Supporting personal collections across digital libraries in spatial hypertext," in Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries, ser. JCDL '04. New York, NY, USA: ACM, 2004, pp. 358-367

[12] Perkins, T., Murphy R. Opportunistic Active and Mediated Cooperation Between a UAV and UGV. International Symposium on Safety Security and Rescue Robotics, 2013, Under Submission

[13] Duckworth, D. Shrewsbury, B. Run the Packbot Backward. A Lesson Learned at a Radiological Exercise. International Symposium on Safety Security and Rescue Robotics, 2013, to appear.

[14] Duncan. B Muphy, R. Data Paths of Context data to IC. Technologies for Homeland Security 2013. Under Submission

[15] Bagedit https://bitbucket.org/daniel_dube/bagedit/wiki/Home