Effects of Alarms on Control of Robot Teams

Shih-Yi Chien, Huadong Wang, Michael Lewis
School of Information Sciences
University of Pittsburgh
Pittsburgh, PA 15260 U.S.A.
shc56@pitt.edu, huw16@pitt.edu,
ml@sis.pitt.edu

Siddharth Mehrotra, Katia Sycara
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15260 U.S.A.
siddharthmehrotra@cmu.edu,
katia@cs.cmu.edu

Annunciator driven supervisory control (ADSC) is a widely used technique for directing human attention to control systems otherwise beyond their capabilities. ADSC requires associating abnormal parameter values with alarms in such a way that operator attention can be directed toward the involved subsystems or conditions. This is hard to achieve in multirobot control because it is difficult to distinguish abnormal conditions for states of a robot team. For largely independent tasks such as foraging, however, self-reflection can serve as a basis for alerting the operator to abnormalities of individual robots. While the search for targets remains unalarmed the resulting system approximates ADSC. The described experiment compares a control condition in which operators perform a multirobot urban search and rescue (USAR) task without alarms with ADSC (freely annunciated) and with a decision aid that limits operator workload by showing only the top alarm. No differences were found in area searched or victims found, however, operators in the freely annunciated condition were faster in detecting both the annunciated failures and victims entering their cameras' fields of view.

INTRODUCTION

Over the past 50 years annunciator driven supervisory control (ADSC) has evolved as the standard solution for human control over systems too complex for humans to monitor. In ADSC a complex system is analyzed to identify parameter set points that, if exceeded, would be an indication of an off-normal state and diagnostic with respect to the subsystem(s) involved. The set point is typically alarmed, through a flashing legend tile indicating the parameter, set point and system involved. The alarms serve to direct the operator's attention from the complexity of the entire system which exceeds human cognitive capacity to the particular subsystem and condition needing to be scrutinized. By reducing the scope of what the operator must consider, ADSC brings the problem back within the operator's cognitive capacity enabling her to control a system that she could not otherwise direct. ADSC essentially converts a complex dynamic control problem into a queuing system in which discrete jobs (alarmed problems) are presented to an operator (server).

The effectiveness of ADSC depends on two things: 1) the decomposability of the system into relatively independent subsystems that can be considered in isolation and 2) the definition of set points that can reliably cue the operator to developments requiring human attention. In conventional applications such as nuclear power plants, refineries, or chemical reactors the lack of independence among subsystems has caused the greatest difficulty. Although ADSC has successfully managed complex systems throughout the world for over half a century, accidents such as the loss of cooling at Three Mile Island (IEEE, 1979) still occur because of the tunnel vision the technique promotes. Where interaction across subsystem boundaries is involved, ADSC leads operators to conclude that things have gone haywire without providing any idea of where to start unraveling the problem. Thrown unexpectedly into dealing with the full complexity of the systems under their direction operators have neither the experience nor intuitions needed to perform their role. Proposed solutions such as encouraging operators to reason about system evolutions at multiple levels of abstraction (Rasmussen, 1986; Vicente, 2002) or displaying global system state (O'Hara, Higgins, & Kramer, 1996) are not widespread because of the complexity of the problems and the difficulty in analysis of considering the potential failure modes.

While applying ADSC to complex industrial processes presents formidable challenges, multirobot control is significantly more difficult because of the difficulty in defining set points. While industrial processes can have clear ranges over which we expect parameters to vary, robot teams do not. For mobile robots the geographical region where they are deployed, separation (dependent on obstacles), execution times (dependent on terrain), etc. all will vary from mission to mission making constant set points infeasible. The alternative of making set points dependent on mission, terrain, adversary, etc. is also infeasible because of the complexity of re-analyzing/re-specifying alarm parameters for every mission.

Research in robot self-reflection (Scheutz & Kramer, 2007) has progressed to the point that it is reasonable to presume robots capable of reliably reporting their own off normal conditions such as an inability to move, unsafe attitude, or other failure inferable from sensed data. For tasks such as foraging in which robots operate with relative independence these individual reports could provide a basis for alarms focusing operator attention on robot(s) in need of interaction.

Because of the complexity of the operator's task in identifying robots in need of assistance and choosing among them, a decision aid assisting the operator in the choice of which robot to service next might decrease mental workload and improve performance. Direction of operators at this level

Figure 2. Alarm condition

of specificity, however, has often met with resistance (Kirlik, 1993). Fielded ADSC systems typically leave the selection of problems from those alarmed to the operator or provide a priori prioritization scheme distinguishing between major and minor problems. Despite early results suggesting that dictating the choice of robot to control may fare poorly in human-machine systems (Crandall et al., 2010), the potential gains warrant further investigation and development of effective means for conveying recommendations.

The present study addresses these issues by comparing a control condition in which operators perform a multirobot urban search and rescue (USAR) task without alarms with an ADSC condition in which self-reported faults are freely annunciated and a decision aid showing only the highest priority alarm.

Method

USARSim and MrCS

The experiment reported in this paper was conducted using the USARSim robotic simulation with 6 simulated Pioneer P3-AT robots performing Urban Search and Rescue (USAR) foraging tasks. USARSim is a high-fidelity simulation of urban search and rescue (USAR) robots and environments developed as a research tool for the study of human-robot interaction (HRI) and multi-robot coordination. USARSim supports HRI by accurately rendering user interface elements (particularly camera video), accurately representing robot automation and behavior, and accurately representing the remote environment that links the operator's awareness with the robot's behaviors. Other sensors including sonar and audio are also accurately modeled. Many validation studies have shown close agreement in behavior and sensing between USARSim models and the robots being modeled.

MrCS (Multi-robot Control System), a multi-robot communications and control infrastructure with accompanying user interface, developed for experiments in multirobot control and RoboCup competition (Balakirsky et al., 2007) was used in this experiment. MrCS provides facilities for starting and controlling robots in the simulation, displaying multiple camera and laser output, and supporting inter-robot communication through Machinetta which is a distributed multi-agent coordination infrastructure.

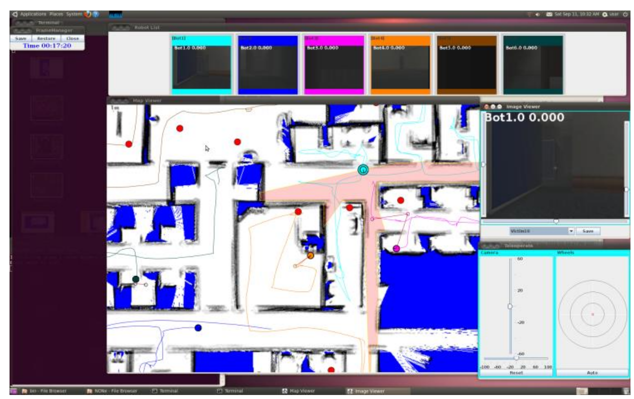

Figure1. Control condition display

Figure 1 shows the control condition with the elements of the conventional MrCS. The operator selects the robot to be controlled from the colored thumbnails at the top of the screen. To view more of the selected scene shown in the large video window the operator uses pan/tilt sliders to control the camera. The current locations and paths of the robots are shown on the Map Data Viewer (bottom left). Robots are tasked manually by assigning waypoints on the map or through a teleoperation widget (bottom right).

The two experimental displays augment the standard MrCS with alarms resulting from simulated robot self-reflection.

1) Alarm (ADSC): The team status window shows each robot's current condition in different colors and briefly summarizes it. Green color indicates the robot is in autonomous condition, yellow shows the robot is in an abnormal condition, such as stuck at a corner or flipped, and when a robot is manually controlled its tile turns white.

2) Decision Aid: This display shows the highest priority alarm for a robot in an abnormal state. Additional alarms can only be reviewed after the presenting problem is resolved.

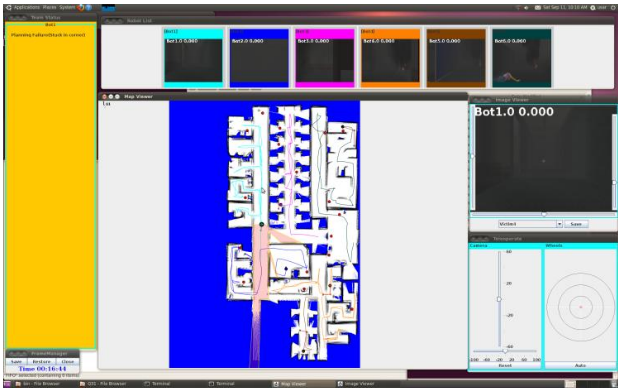

Figure 3. Decision Aid condition display

USAR Foraging task

When an operator detects a victim in a thumbnail a complex sequence of actions is initiated. The operator first needs to identify the robot and select it to see the camera view in a larger window and to gain the ability to stop or teleoperate the robot. After the user has successfully selected a robot, it must be located on the map by matching the window border color or numerical label. Next the operator must determine the orientation of the robot and its camera using cues such as prior direction of motion and matching landmarks between camera and map views. To gain this information the operator may choose to teleoperate the selected robot to locate it on the map, determine its orientation through observing the direction of movement, or simply to get a better viewing angle. The operator must then estimate the location on the map corresponding to the victim in the camera view. If "another" victim is marked nearby the operator must decide whether the victim she is preparing to mark has already been recorded on the map.

Detecting and restoring a failed robot follows a similar time course with the act of teleoperating the rescued robot to the next waypoint substituting for marking the victim.

Experimental Conditions

A large USAR environment previously used in the 2010 RoboCup Rescue Virtual Robots competition (Robocup Rescue VR, 2010) was selected for use in the experiment. The environment was an office like hall with many rooms and full of obstacles like chairs, desks, and bricks. Victims were evenly distributed within the environment. Maps were rotated by 90^{o} and robots entered the environment from different locations on each of the three trials. Because the laser map is built up slowly as the environment is explored and the office like environment provides few distinctive landmarks there was little opportunity for participants to benefit from prior exposure to the environment. Robots followed fixed paths from each set of entry points simulating the autonomous navigation used in earlier (Chien, Wang, & Lewis, 2010) studies. The map contained 20 points at which failures were injected. Upon reaching a failure point the robot experienced a failure, such as becoming entangled with a chair. The operator then needed to assume manual control to teleoperate the robot out of its predicament to its next waypoint. The experiment followed a three condition repeated measures design comparing the conventional MrCS displays with MrCS augmented by alarm panels. Conditions were fully counterbalanced for Map/starting points and display with 5 participants run in each of the six cells

Participants and Procedure

31 paid participants were recruited from the University of Pittsburgh community balanced among conditions for gender. None had prior experience with robot control although most were frequent computer users. Due to a system crash data was lost for one participant.

After providing demographic data and completing a perspective taking test, participants read standard instructions on how to control robots via MrCS. In the following 15 minute training session, participants practiced control operations. Participants were encouraged to find and mark at least one victim in the training environment under the guidance of the experimenter. After the training session, participants began the first 15 minute experimental session in which they performed the search task controlling 6 robots in the first assigned condition. At the conclusion of the session participants were asked to complete the NASA-TLX workload survey (Hart & Staveland, 1988). After brief breaks, the next two conditions were run accompanied by repeated workload surveys.

RESULTS

Data were analyzed using a repeated measures ANOVA comparing search and rescue performance between the control and the two alarmed displays. No difference was found on the overall performance measures areas covered ($F_{1,29} = .488$, $p = .490$), victims found ($F_{1,29} = .294$, $p = .592$), or NASA-TLX workload survey ($F_{1,29} = 2.557$, $p = .121$). Significant effects were found on measures relating to operator strategy and the ways they performed their tasks.

Neglect times

The Neglect Tolerance model (Crandall et al., 2005) holds that increasing robots' autonomy allows robots to be neglected for longer periods of time making it possible for an operator to control more robots. Neglect time, therefore, can be considered an indirect measure of operator efficiency. Robots in the Decision Aid condition were neglected longer than in the Control condition ($p = .033$, $SD = 619.507$) but did not differ significantly from the Alarm condition. The neglect times were Alarm = 1741, Decision Aid = 1887, and Control = 1629 seconds.

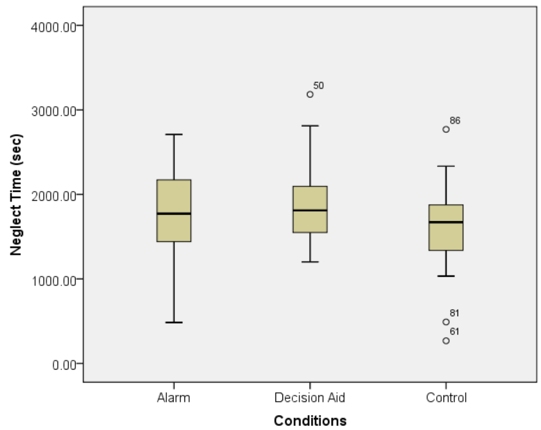

Figure 4. Neglect Time

Fault Detection Time

Fault Detection time was defined as the interval between the initiating failure and the selection of the robot involved in that event.

Cumulative Fault Detection times were significantly shorter for participants in the Alarm condition, p = .021, with a cumulative Fault Detection time of 933 seconds. Times for Decision Aid and Control conditions were 1120, and 1210 seconds respectively. A pairwise T-test shows a significant difference between Alarm and Control conditions (p = .021, SD = 607.914).

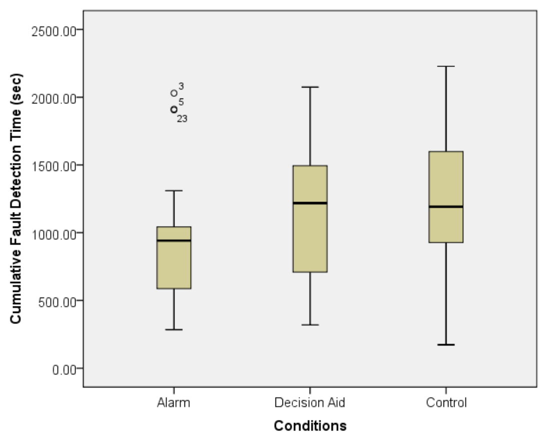

Figure 5. Cumulative Fault Detection time

Average Fault Detection times show a similar advantage for the Alarm condition for noticing robots in trouble, p = .014, SD = 76.583. These waiting times were Alarm 90, Decision Aid 110, and Control 128 seconds.

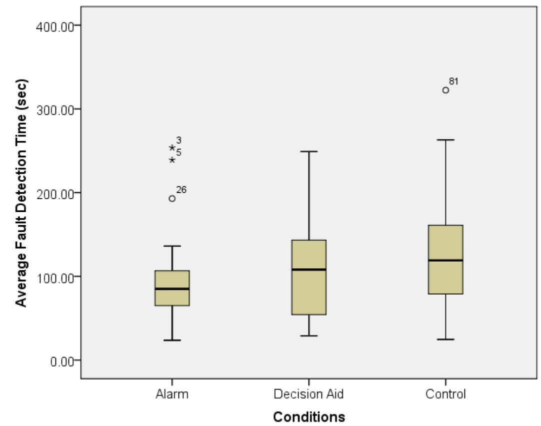

Figure 6. Average Fault Detection Time

Victim Delay time was defined as the interval between when a victim first appeared in a robot's camera and the selection of that robot. Victim Delay time again differed across conditions with average times of Alarm 1303, Decision Aid 1548, and Control 1559 seconds. A pairwise T-test shows differences between Alarm and Decision Aid (p= .041, SD = 613.725), and Alarm and Control conditions (p = .025, SD = 578.945).

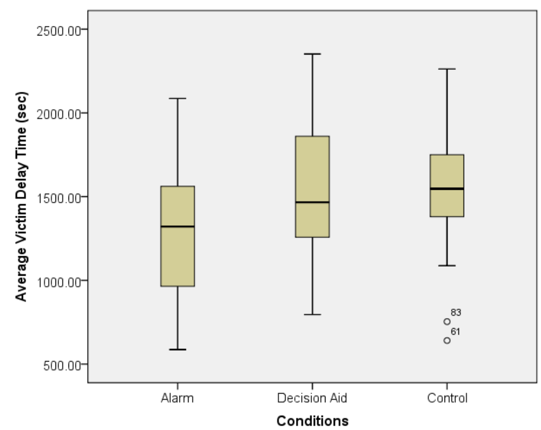

Figure 7. Average Victim Delay Time

Select to Mark per victim

A related measure, Select-to-Mark, is defined by the interval between selecting a robot with a victim in view and marking that victim on the map by the process described earlier. Select to mark times can be interpreted as a measure of situation awareness (SA) because they require the operator to orient and interpret the environment.

For this measure the results are reversed with users in the Alarm condition taking the longest times (17.56 sec) and the Control the shortest (14.91 sec) with the Decision Aid condition (16 sec) again falling in between. There was no overalleffect forselect to mark time acrossthe three experimental conditions (F(1.669,56) = 1.618, p = .212). A pairwise T-test, however, shows a significant difference between Alarm and Control conditions (p = .025, SD = 6.02).

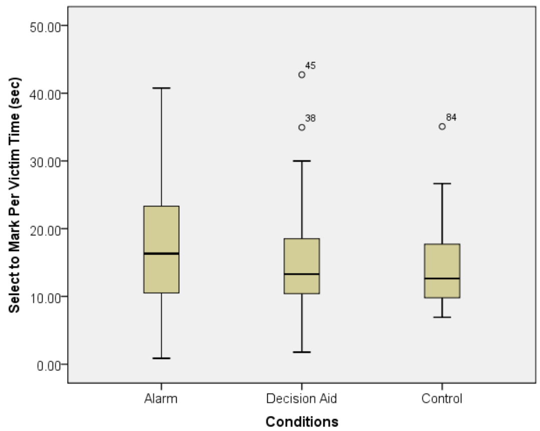

Figure 8. Select to Mark Per Victim Time

DISCUSSION

Allowing robots to alert operators to their abnormal states not only reduced the need to monitor for failures as evidenced by the reduced Fault Detection times, but appear to have freed cognitive resources to monitor the video feeds for victims leading to reduced Victim Detection times as well. The increased select-to-mark times for operators receiving alarms suggest that operator strategies may have changed in ways that shift attention from the map to the alarm panel and thumbnails forcing them to reacquire SA before marking victims on the map. An alternate explanation may be that the more complete information available to Alarm condition operators allows them to consider problems in parallel interleaving planning for further interactions with the marking task.

While the information provided by the Alarm and Decision Aid displays should be equivalent for an servicing robots in a sequential fashion this was not the case for our data with Decision Aid conditions falling somewhere between the Alarm and Control conditions. Anecdotal observations suggest that this may be because some participants chose to ignore the Decision Aid preferring to control robots from the thumbnails and map as in the control condition. User acceptance of highly prescriptive decision aids has been a longstanding problem in human-machine systems (Kirlik, 1993). Whether it is an aversion to being controlled by a machine or cognitive dissonance from being unable to understand the basis for a machine's decisions prescriptive aids are often simply turned off. This is especially relevant to multirobot control for the types of tasks studied here because most of the technical assistance we could provide involves using sophisticated scheduling models to help the operator choose the right robot to control. While the simple Decision Aid used in this study did not offer the advantages of sophisticated priority queues (Crandall et al., 2010) or models based on service differentiation (Xu et al., 2010) we can never realize advantages from more sophisticated aiding, unless we can convey this guidance to the operator in a more effective way. Similar results were reported by (Crandall et al., 2010) for queue driven multirobot control that dictated decisions. Since the ADSC Alarm display is already showing advantages over the basic system, modifying it to "suggest" rather than "dictate" the next robot may be a way to improve performance without alienating the operator.

ACKNOWLEDGMENT

This research has been sponsored in part by AFOSR FA955008-10356 and ONR Grant N0001409-10680

REFERENCES

Balakirsky, S., Carpin, S., Kleiner, A., Lewis, M., Visser, A., Wang, J. & Zipara, V. (2007). Toward hetereogeneous robot teams for disaster mitigation: Results and performance metrics from RoboCup Rescue, Journal of Field Robotics.

Cao, Y., Fukunaga, A., & Kahng, A. (1997). Cooperative mobile robotics: Antecedents and directions, Autonomous Robots, 4, pp. 1-23.

Chien, S., Wang, H, & Lewis, M. (2010). Human vs. algorithmic path planning for search and rescue by robot teams, Proceedings of the 54th Annual Meeting of the Human Factors and Ergonomics Society (HFES'10).

Crandall, J. & Cummings, M. (2007). Developing performance metrics for the supervisory control of multiple robots, ACM Conference on Human Robot Interaction (HRI'07).

Crandall, J., Cummings, M., Della Penna, M. & deJong. P. (2010). Computing the Effects of Operator Attention Allocation in Human Control of Multiple Robots. IEEE Transactions on Systems, Man, and Cybernetics -- Part-A.

Crandall, J., Goodrich, M., Olsen, D. & Nielsen, C. (2005). Validating human-robot interaction schemes in multitasking environments. IEEE Transactions on Systems, Man, and Cybernetics,.

Hart, S. & Staveland, L. (1988). Development of a multi-dimensional workload rating scale: Results of empirical and theoretical research. In P. A. Hancock & N. Meshkati (Eds.). Human mental workload, Amsterdam, The Netherlands: Elsevier, pp. 139-183.

IEEE (1979). Special issue on Three Mile Island accident and the nuclear power industry in the United States. IEEE Spectrum, 16(11).

Kirlik, A. (1993). Modeling Strategic Behavior in Human-Automation Interaction: Why an "Aid" Can (and Should) Go Unused, Human Factors: The Journal of the Human Factors and Ergonomics Society, 35(2), pp. 221-242.

O'Hara, J., Higgins, J. & Kramer, J. (1996). Advanced information systems design: Technical basis and human factors review guidance, NUREG/CR-6633, US NRC.

Olsen Jr., D.R. & Wood, S. B. (2004). Fan-out: Measuring Human Control of Multiple Robots. Human Factors in Computing Systems (CHI'04), Vienna, Austria: ACM Press.

Parasuraman, R., Barnes, M. & Cosenzo, K. (2007). Adaptive automation for human-robot teaming in future command and control systems, International C2 Journal, 1(2).

Rasmussen, J. (1986). Information Processing and Human-machine Interaction: An Approach to Cognitive Engineering.

Robocup Rescue VR. (2010). Retrieved September 15, 2010, from http://www.robocuprescue.org/wiki/index.php?title=VRCompetitions#Singapore_2010

Savla, K., Nehme, C., Temple, T. & Frazzoli, E. (2008). On efficient cooperative strategies between humans and UAVs in a dynamic environment, AIAA Guidance, Navigation and Control Conference.

Scheutz, M. & Kramer, J. (2007). Reflection and reasoning mechanisms for failure detection and recovery in a distributed robotic architecture for complex robots, Proceedings of the 2007 IEEE International Conference on Robotics and Automation.

Vicente, K. J. (2002). Ecological Interface Design: Progress and challenges. Human Factors, 44, pp. 62-78.

Xu, Y., Dai, T., Sycara, K., & Lewis, M. (2010). Service level differentiation in multi-robots control, International Conference on Intelligent Robots and Systems (IROS'10), IEEE.