

A Hands-Off, Multi-Robot Display for Communicating Situation Awareness to Operators

Jacob Crossman* and Robert Marinier

Soar Technology, Inc.
Ann Arbor, MI, USA
jcrossman@soartech.com
bob.marinier@soartech.com

Edwin B. Olson

Dept. of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, MI, USA
ebolson@umich.edu

Abstract—In this paper we introduce work toward this goal of enabling an operator to effectively maintain situation awareness over a team of heterogeneous robots. Most existing operator control units (OCUs) are designed for tele-operation of a single robot and require the constant attention of the operator. As the role of robots increase in important fields such as bomb disposal; intelligence, surveillance, and reconnaissance (ISR); and search and rescue it becomes critical to improve this ratio by enabling multiple robots to be monitored and controlled by a single operator. Toward this goal we have developed the Situations, Actions, Goals, and Environment (SAGE) interface, which utilizes numerous techniques for maintaining operator situation awareness by aiding the operator’s perception, comprehension, and projection when controlling a team of ground robots. Techniques that we developed include dynamic view control, event detection and prioritization, uncertainty display, view-invariant markers to make state information globally visible, and animations to avoid jarring distractions. SAGE’s interface was automated making it completely hands off, thus the operator could save interaction time for critical tasks such robot tasking. SAGE was deployed by Team Michigan as part of its OCU suite in the 2010 Multi-Autonomous Ground-robotic International Challenge (MAGIC). Team Michigan won this competition in part because of its innovative OCU designs, including SAGE, that enabled two operators to control 14 robots simultaneously – a 1-to-7 operator to robot ratio, higher than any other team.

Keywords—situation awareness; robot teams; operator control unit; adaptive user interface; event detection; MAGIC

I. INTRODUCTION

This paper discusses work toward solving a fundamental problem in current ground robot operations – the large amount of attention and interaction required to control a team of unmanned vehicles. This work was accomplished as part of our winning effort in the Multi-Autonomous Ground-robotic International Challenge (MAGIC) [1][2][3]. The competition presented each team with the challenge of directing a team of semi-autonomous ground robots to map a large space (indoor and out) while looking for and disabling “dangerous” objects. This task was designed to mimic the type of intelligence, surveillance, and reconnaissance (ISR) missions typically presented to small military units. Each team had to deploy at least three robots (our team deployed fourteen) and was allowed only two human operators. Operators were penalized for interacting with any part of their system, thus a premium was placed

on autonomy – not just of the robots, but of the operator control units as well. Thus the competition provided an excellent framework within which to research and test ideas for minimizing user-robot interaction while maintaining or increasing operator situation awareness of high-level tasks.

Today’s commercial ground robots, with a few exceptions, are tele-operated using an operator control unit (OCU). Each robot is controlled by a single person; that is, there is an operator to robot control ratio of 1-to-1. Typical OCUs present the user with one or more video feeds from cameras mounted on the vehicle, which are augmented with details about the robot’s control state and pose (Figure 1). The operator typically uses a joystick-type controller to activate the various motors on the robot to move it and any mounted manipulators.

Controlling a robot through this type of OCU requires constant, full attention by the operator. Even with this attention, the operator can easily become confused [4] due to difficulties with depth perception, independent movement of camera and vehicle, poor map design, and cluttered displays.

Despite these challenges ground robots have proven to be invaluable for many tasks, especially those in environments that pose danger or are inaccessible to people. For example, ground robots are now commonly used to dispose of roadside bombs in war zones, search buildings for barricaded crime suspects, and search for signs of life in disaster zones. Thus there is an increasing demand to decrease the operator-to-robot ratio to enable tasks such as search and rescue and surveillance to be accomplished more efficiently. Indeed, especially for less dangerous tasks, robot-human teams may only make sense if there are multiple robots per operator, since each robot is, by itself, less capable than a human, and thus one human per robot does not gain much.

Solutions to this problem require two fundamental capabilities. The first is *autonomy*, which is required to reduce operator interaction time. For an operator to leave a robot alone the robot must be able to maneuver and act on its own at least part of the time. In MAGIC, our team of robots was able to execute many behaviors, such as path planning and movement, autonomously. As we will show later, autonomy need not be limited to the robots themselves, but can also be applied to the OCU itself to minimize the interaction time needed to maintain situation awareness.

* Corresponding author



Figure 1. Talon OCU [5]

The second problem, and the principle focus of this paper, is real-time, task-relevant *situation awareness (SA)*, which is required to increase allowable operator neglect time. Even when a robot is executing an autonomous mission the operator must still spend significant time monitoring the robot to ensure (1) that it does

not encounter a situation that it cannot handle, and (2) that opportunities it encounters (e.g. a target moving across its field of view) are exploited. To achieve low operator-to-robot ratios such as the 1-to-7 ratio we demonstrated in MAGIC, the operator must be able to shift attention as the important aspects of the task change. Thus an operator needs to be able to ignore robots for significant periods of time, while simultaneously maintaining global SA and responding to critical events.

II. PRIOR WORK

As mentioned above, the current operational standard in robot control and situation awareness is tele-operation via an operator control unit (OCU). Figure 1 shows a deployed OCU for the Talon UGV. Another example with similar low-level data is the Wayfarer research OCU for the Packbot UGV [6]. As with most ground OCUs these exemplars are centered on a first-person camera view from the UV. Detailed information about the vehicle pose and control status are provided in the perimeter. Many users find these sorts of OCUs overwhelming given the detailed information and control widgets.

Research has shown that these types of OCUs are difficult for operators in large part because the operator has difficulty maintaining situation awareness even though their attention is completely focused on the UV. For example, in experiments reported in [4], operators spent an average of 30% of each run “acquiring SA while no other task was being done. Despite this time spent trying to acquire SA, users often expressed confusion about where their UVs were located ...” There are many reasons for this difficulty including the independent movement of camera and vehicle, difficulty perceiving depth and distances via a camera, and poor overhead map design.

Furthermore, when OCU displays are cluttered it is difficult for the operator to pick out the specific information that is needed at any moment. Thus one thread of research (including some of our own work [7]) has focused on providing information via multiple modalities such as voice or force feedback.

Recently, work has been done to develop more streamlined OCUs, including those that support maintaining SA over multiple robots. Multi-robot OCU (MOCU) is an architecture designed to make it easy to incorporate customized displays and controls for many heterogeneous platforms and missions in a single interface, allowing for simultaneous control of multiple robots [8]. MOCU includes a monitor mode (Figure 2), which

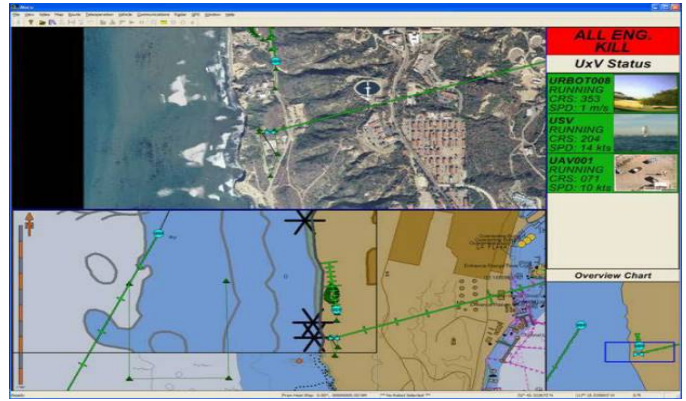


Figure 2. Figure 1: MOCU interface

is most similar to the work described here. In monitor mode, MOCU shows an overhead map as well as first-person views from each robot, allowing the operator to maintain better SA. However, MOCU’s focus is in providing an architecture in which OCUs can be developed, as opposed to being an OCU itself. Thus, it may be possible to create type of display described in this paper in MOCU, but no such MOCU configuration currently exists.

NIFTi [9] is another OCU designed to help a user maintain SA over multiple robots. NIFTi provides a 3D map of the environment and first-person views from the robots. Both MOCU and NIFTi seem primarily focused on making robot control easier, with SA itself provided by a map and camera feeds. Neither attempts to actually detect and prioritize events for the user or automatically control the display to emphasize important aspects of the situation and keep distractions to a minimum, and neither attempts to capture and display uncertainty.

While existing OCUs are still fairly rudimentary, significant work has been done at the theoretical level. In particular, Endsley and others [10][11] have identified several factors that should play a role in OCU design. At a high level these include the three levels of SA: Perception of elements in the environment (level 1), comprehension of the current situation (level 2), and projection of future status (level 3). As we will show, our work specifically targets situation awareness aiding the operator in maintaining these levels of awareness for a mission that includes many ground robots.

III. OUR APPROACH: THE SAGE INTERFACE

In response to these challenges we developed the Situations, Actions, Goals and Environment (SAGE) user interface and deployed it as one of four displays used by two robot operators in the MAGIC competition [12]. Our goal with SAGE was to increase operator and competition judge situation awareness (Table 1) while minimizing operator interaction with the system. Our solution included these key innovations:

1. A mission-oriented global display that presented only critical information to the user and continuously re-oriented to keep relevant spatial information in view
2. Automatic event detection and highlighting to alert the operator to situations that required attention
3. Uncertainty tracking and visualization

In addition to these capabilities, we challenged ourselves to design an interface that was completely hands-free. Though the operators had to occasionally interact with other interfaces to send robots commands or correct errors in the global map, the SAGE interface operated autonomously.

A. The Architecture

A significant part of SAGE’s value was derived from the processing it did internally, in addition to its specially designed visualizations. To enable autonomous display control SAGE was required to build and maintain its own internal SA, which it could then communicate to the user in pieces as they became relevant to operator’s activities.

Figure 3 shows a high-level view of the SAGE software architecture. SAGE was integrated into a larger, distributed control system from which SAGE obtained all of its data regarding mission state and activities. SAGE’s three main functional layers were as follows:

- Data filtering, aggregation and fusion
- Autonomous display control
- 3D Visualization

Data filtering, aggregation and fusion created a tractable data model for use by the other processes in the system. The complex fusion processes were implemented by distributed components dedicated specifically for that task. For example, the critical global map was generated by a separate component implementing global simultaneous location and mapping (SLAM) functions [2][3]. These processes communicated their output data to SAGE and other components through the team’s Lightweight Communications and Marshaling (LCM) messaging infrastructure [13].

Autonomous display control created, configured, managed and destroyed display components based on the current context. The heart of this layer was a viewport manager that managed the highlight views that appeared automatically on the right side of the SAGE display. An animation manager was used to smoothly animate most state changes within the GUI and environment thus enabling the user to retain context as focus shifted through the mission.

All visualizations were rendered in 3D using Java’s JOGL APIs. Terrain and walls were rendered using custom primitives and geometry management to maintain high refresh rates for large terrain sizes. Going into the competition we were uncer-

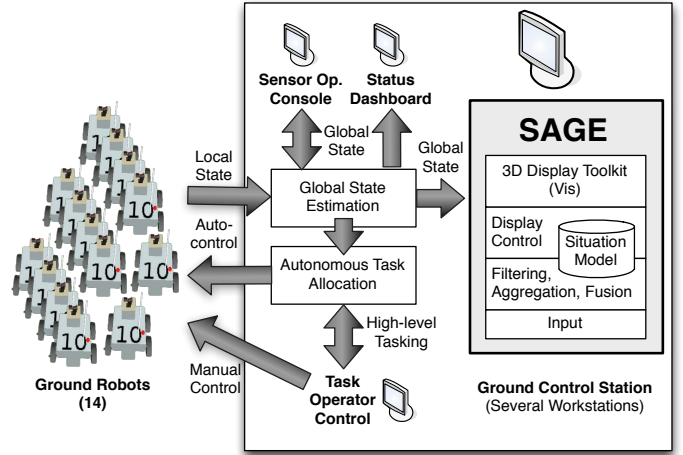


Figure 3. The SAGE architecture

tain of the size of each stage and estimated from rough sketches that they could be as large as 500m x 500m. Given a grid size of 0.1m SAGE potentially needed to maintain and render the information from 25million cells. We built SAGE to handle this scale, though in practice the typical sizes encountered during the competition were closer to 2.5 million cells due to unused spaces in the competition area (see Table 2 in Section IV for related results).

B. The Overall Layout Design

The overall layout is designed to balance global situation awareness (e.g., where all the robots are) with local situation awareness (e.g., what is happening at a specific location). By default, the display shows the global situation, automatically adjusting to keep all of the action in view (see section C below). However, when important events are detected, side panels dynamically pop up to emphasize the event (see section D below). Overall, we reserve at least half of the screen for global situation awareness, so even when many events are happening, the users do not lose global context. Figure 4 shows SAGE in action.

C. Global Situation Awareness

Our goal for the main SAGE display was to present the user with all information relevant to the global mission, and only that information. Thus rather than start with the data the robots produce and figuring out how to display it, we started with the needs of the user and selected only data elements that matched these needs. These needs included the following:

1. The location, identity, orientation, and status of all active robots (robots were sometimes deactivated or used for special purposes like communications relays). This supports Level 1 and Level 2 SA.
2. The current task being executed by the robot. This supports Level 2 and Level 3 SA.
3. The location and perceived status of important objects (e.g. OOs – Objects Of Interest). This supports Level 1 and Level 2 SA.

To address the first information need, we built a display control algorithm that continuously monitors both the robot

TABLE I. SAGE FEATURES AND HOW THEY SUPPORT LEVELS OF SA

Level 1: Perception	Dynamic view changes Dynamic robot and object markers Dynamic event notifications Duplicate event detection Old event fading Fog of war
Level 2: Comprehension	Robot animations Task animations Event notifications Event prioritizations Old event fading Uncertainty display
Level 3: Projection	Robot goal animations Imminent problem notification



Figure 4. SAGE screen shot showing overhead view (left) and two events of different priorities (right). Also visible: robot markers, disabled robot, mobile and stationary objects.

position and state and automatically adjusts its view to fit these robots. SAGE maintained a set of all active robots for which it maintained an axis aligned bounding box. It then computed view frustum parameters that fit a top down, isometric ("god's eye") view (Figure 5, left) to this bounding box. To keep the user from becoming disoriented, the system fixed the view angle and direction (thus the axis alignment), essentially moving the camera back and forth along a vector passing through the camera and the point within the bounding box that passes through the center of the screen.

An effect of this design is that, as the robots spread out and the view position moves backward, the human eye can no longer see critical information (including the robot position for particularly wide spreads). To solve this problem we designed view-invariant markers that were rendered underneath the robots (Figure 5, right). These markers were high contrast (black and white) and easily visible. They give critical information at a glance such as robot position, robot identity, robot orientation, robot type, and robot status. When zoomed out, these markers stay a fixed size. When zoomed in they gradually disappear under the robot 3D model, allowing the user to see and focus on the 3D robot model, which contains the same infor-

mation but in more realistic form. The use of highly visible markers and automatic adjustment of the visible area supports the user's perception of information as well as comprehension of various aspects of each robot.

To address the second information need we implemented subtle task animations. The goal was to make it easy to see what the robots were doing, while not distracting the user. The most visible of these animations is the navigation task animations where a line is used to connect a robot to its target area, depicted as a small circle. The color of the line emphasizes the type of robot, and thus whether this is an exploration task or a neutralizing task; neutralizing robots are shown in a brighter color since they are generally more important for the operator to attend to. A dot is animated along the line to show the direction of movement. When the dot reaches the end of the route, a small ring expands highlighting the target area. The display of each robot's task supports user comprehension of what they are doing, while the display of the robots' goals helps the user project where they are likely to be in the near future.

Another more subtle animation, but just as important, is the animation of robots between their reported positions. In the MAGIC competition, robot poses were typically reported 1-2 times per second, although the rate varied depending on a number of factors. Even at this rate, simply "teleporting" the robot markers to their new locations and orientations would be a jarring experience, making it difficult for users to track individual robots in many cases (e.g., when many robots are close together), a problem with Level 2 SA. Our solution was to smoothly animate robot markers from their previous pose to their new pose. The animation manager attempted to automatically adjust the animation rate to make continuously moving robots appear to move continuously, and thus the robot markers tended to be about one update behind the actual robot positions.

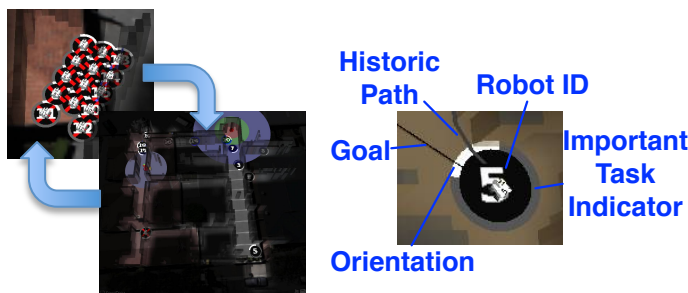


Figure 5. (Left) Automatic zoom control to keep robots in view. (Right) Robot marker.

However, given the update rate and the high level of autonomy of the robots, this tradeoff was more than justified given the improvement in usability.

Finally, to address the third information need, we developed 3D models and markers for of the key objects. We used a color scheme to draw attention to risky objects (e.g. OOIs) and downplay less important objects (e.g. doors). Additionally, we rendered the OOIs together with richly saturated range rings so the user could see dangerous situations at a glance (robots were required to get within a certain distance of stationary OOIs to neutralize them but would be disabled if they got too close). The system automatically removed these rings when an OOI was neutralized. This automatic management of the visibility of information helps the user focus on important information while also providing critical context.

D. Event Detection and Highlighting

One particularly difficult challenge to building an autonomous display was the detection and automatic display of key user information. In a sense, the system must "know" what the user needs at a particular point in the mission and display it in the appropriate way.

Solutions to this problem can be enormously complex incorporating user models, state tracking and anticipation, and complex information models. We chose to implement a simple, but effective design that fit both the development time and robustness requirements of the competition.

At the core of this design is an event detection system. Through design analysis and discussions within the team, we defined a set of event classes that we understood to be important to a user; for example, a robot sensing an OOI for the first time. In more sophisticated systems, this information might be represented as a formal user model, but SAGE represented it as hard coded event classes for expediency.

For each event class we designed event detectors. Where possible, these event detectors leveraged already fused and/or processed data that the distributed robots were producing. For example, robot task state was readily available through messages published by the tasking GUI. Other events had to be derived from sensed data; for example, a civilian too near to an OOI. This event had to be deduced from the global map and object reports from the robots.

For each event class we pre-defined a view mode appropriate for that event (Figure 6). This view was then instantiated as a side panel, with the view automatically reconfiguring (using smooth animation) as events occur and fade. This view management system provided the user with a contextually relevant view that gave them SA rapidly. For example, when new objects were discovered, the system automatically constructed an isometric view that showed the object and its surrounding context to help the user see nearby obstacles, robots, etc. However, when a robot switches to the neutralization task, SAGE creates a much closer third person view behind and slightly above the robot, giving the operation a picture of the immediate obstacles and the robot's field of view.

When a new event side panel is added, it smoothly animates into position, with the view zooming in from a distance into the

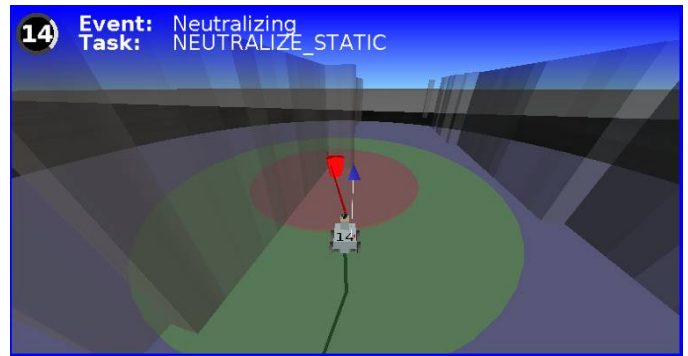


Figure 6. A side panel showing a close-up of a neutralizing event. The robot, task, and surrounding context are clearly visible.

proper view for the event (Figure 7). Simultaneously, existing panels smoothly animate into their new positions. These animations are more than eye candy; the smooth animations make it much easier for an observer to keep track of where things are as they move around, and the zoom effect makes it easier to see where in the global context the event is taking place.

To prevent view clutter and to maintain visual consistency we implemented a view priority system and rules for view configuration. Higher priority events were displayed at the top of the screen. New events that had the same priority as existing events were added at the bottom of the view thus keeping the position of equal priority events consistent. Additionally, we only allowed two of low priority events in the side, but up to four medium or high priority events. This prevented the side panels from distracting the user with too much unimportant information, but showed the user as much important information as possible without compromising the rest of the display (see Table IV in Section IV for related results).

Finally, we had to address another real-world issue that arose during testing. The MAGIC robots were unable to assign ids to objects they detected (essentially they lacked level 1 fusion algorithms for objects). Thus new object events would often arrive in bursts as the same event was detected multiple times, cluttering the display with essentially duplicate information. Our solution was to have the SAGE GUI use simple spatial rules to combine events into a single user visible event, and then to fade the event over time. Further events of the same type in a geographic region then modulated the fade factor (i.e. kept the event alive longer). Though admittedly ad hoc, this approach was effective in reducing duplicate events (see Table III in Section IV for related results).

The approach described here addressed all levels of SA. The dynamic appearance and coloring of the side panels draws the user's attention, while the content of the panels helps the user understand what important

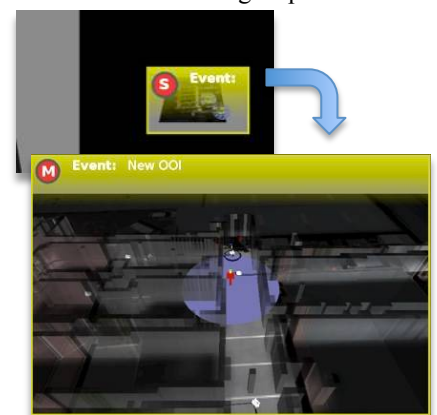


Figure 7. Side panels animate into view

events are happening. For some events, such as when a civilian wanders too close to an OOI, the notification helps the user understand that an undesirable outcome (e.g., a civilian casualty) is imminent.

E. Uncertainty Management and Display

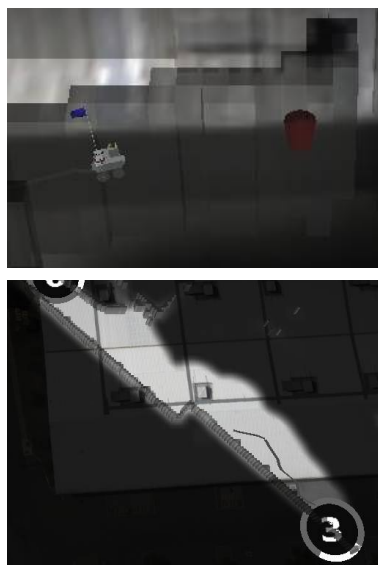


Figure 8. Uncertain walls are rendered translucent (top). Unexplored areas are rendered dark (bottom).

Another challenge facing the interface design was the presentation of information quality to the user. Sensed information varied in quality, being both noisy and, at times, inaccurate. We wanted to build a display that did not mislead a user by displaying as fact what was actually uncertain.

We chose to represent uncertainty in sensed objects as transparency. The more transparent an object appears the less certain the system is about it. This appears in two places. First, walls are drawn with varying levels of transparency based on the confidence that there is actually a wall there (Figure 8, top). This confidence is computed from the global map. Each cell in the SAGE display consists of multiple cells in the global map, and the transparency is computed as a function of occupancy. The more cells in the global map are occupied (i.e. the more positive returns from that region), the less transparent the wall is drawn. This helps a user understand the uncertainty of information for a given area (Level 2 SA). Second, mobile objects are faded over time. Upon detection SAGE shows a mobile object with full opacity because the information is highly certain. It then gradually fades the image over time as the information ages. Thus the user is not erroneously led to believe that a mobile object that was previously detected is still in its old position. This prevents a user from being distracted by obsolete information while also

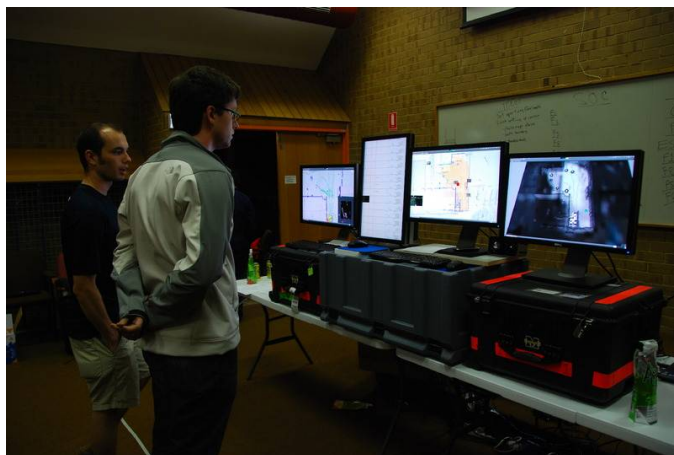


Figure 9. Team Michigan's operators during a practice session. SAGE is the far-right display.

understanding the status of information that is displayed.

SAGE also applied visual cues to show uncertainty due to lack of information (because an area had not been explored). The visualization technique was borrowed from real time strategy game "fog of war" (Figure 8, bottom). The lighting of the ground plane for unexplored regions was modulated to be very dark. Previously sensed regions were rendered with bright lighting. The transitions between these areas were blended using an averaging function, thus transition areas appear as a gradient representing gradually increasing uncertainty. The brightness helps draw the user's attention to areas currently occupied by the robots, while also making it obvious at a glance which areas still need to be explored.

IV. USE IN THE MAGIC INTERNATIONAL ROBOT COMPETITION: RESULTS

The MAGIC competition was held in Adelaide, Australia in October 2010. Team Michigan beat four other finalists to claim first place. In Phase 1, all dangerous objects were identified, without losing any robots, and the area was completely mapped. Phase 2, a slightly larger area, mobile threats disabled two robots, but were neutralized by the others. All but one threat were identified, and the area was completely mapped. Phase 3 was a much larger and more challenging area. Essentially, nearly full operator control was required to navigate the challenges, and no team did well.

In the competition, SAGE was deployed as one of four interfaces in the operator control station (Figure 9 and Figure 10). Each interface was designed to meet a specific function required by the operators. The Sensor Operator Console (SOC) displayed the SLAM graph that is being optimized. An operator corrected map errors via this display. The Task Operator Console (TOC) displayed the positions of the robots and their paths. An operator could override automatic tasking and also introduce constraints (e.g., the red area in Figure 10 is a "no-go" area). The Status Dashboard showed the current status of each robot, including low-framerate video stream (left column), the local map (second column), and other status, including indicators of common problems, battery levels, etc.

Strictly speaking, the operators could perform their tasks without the SAGE display; all required information is present in the other displays. However, the other displays create an information overload problem, where issues and events may not be obvious to the operators. SAGE's value is in presenting the situation in a more user-friendly manner. Once the operator is aware of an issue, he can use the other displays to get more detailed information, or to exert control over the system.

Additionally, our team used an unconventional division of labor. Whereas most teams merely divided the robot team between operators, with each person responsible for all aspects of his part of the team, Team Michigan divided responsibilities based on task. One operator was responsible for maintaining the global map, and thus primarily interacting with the SOC. The other operator was responsible tasking the robots, and thus primarily interacted with the TOC and Status Dashboard. Additionally, a team of judges monitored the operators and the event from positions behind the robot operators.

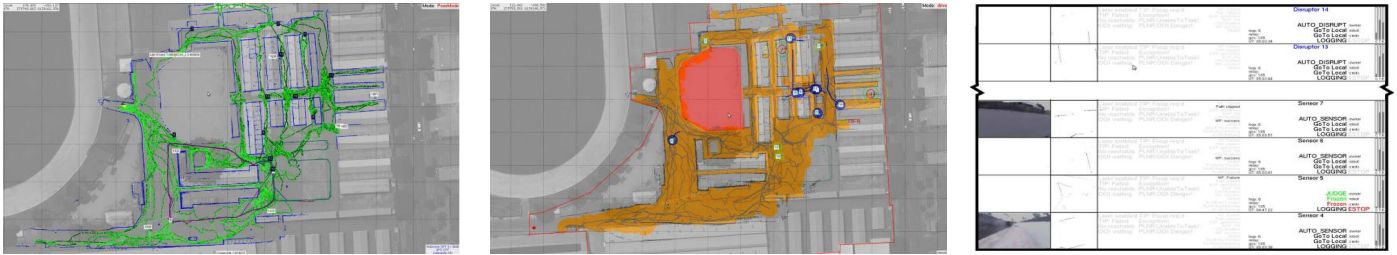


Figure 10. The other operator displays: The Sensor Operator Console (left), the Task Operator Console (center), and the Status Dashboard (right; a subset shown).

SAGE was used for three purposes: by the control operator to maintain global SA, by the judges to understand the overall progress of the mission and the activities of the robots, and by research team to demonstrate and explain the activities of the autonomous robot team to spectators.

Anecdotally speaking, several judges and observers remarked at the unique nature of the display, and some judges were using it in addition to their own judging displays to track the team’s progress. The spectator area had live feeds of all of the operator displays, the judge displays, and cameras. Team Michigan’s leader, Professor Olsen, narrated events, using all the displays. SAGE was not without problems; in particular, the nature of the automatic task assignment often repeatedly assigned and unassigned a task to a robot in rapid succession, resulting in side panels that would pop up and then immediately go away.

While the "one-shot" nature of the competition precluded evaluating the effectiveness of our team both with and without SAGE, both team members and observers found the display to be helpful in maintaining situational awareness. Some of the contest judges, who had access to ground truth data provided by an elaborate system of ultra-wideband tracking devices, found SAGE to be useful. In part, this was due to providing an easily understood depiction of the operating environment. But it also provided an insight into the otherwise unobservable state of the system, showing what threats the robots were aware of and depicting the future action plans for robots.

We recorded data about several aspects of SAGE’s performance, both as a software system and as an operator display.

One aspect of SAGE is its ability to scale to handle large areas (see section III-A). Table II shows the number of map cells reported to SAGE for in each phase. The number of cells fluctuated throughout each phase as the map was expanded and corrected, going up and down. In general, the map only contained the maximum number of cells very briefly, while the final number of cells more accurately represents the stable size that the map grew to over the course of the phase. SAGE was designed with a size of approximately 25 million cells in mind, although during the competition the areas of each phase were much smaller. Overall, SAGE was able to handle the maximum number of cells, corresponding to an area over 500m x 500m.

TABLE II. NUMBER OF CELLS HANDLED BY SAGE

	Phase 1	Phase 2	Phase 3
Max Cells	8,367,618	31,976,766	13,890,250
Final Cells	2,722,552	7,960,896	1,800,755

Another aspect of SAGE is its ability to eliminate duplicate events by merging new events with existing events, and thus avoiding operator distraction with notifications of events that have already been reported (see section III-D). Table III demonstrates the effectiveness of the duplicate event merging process. Not all events created had the potential to suffer from duplicate reporting (e.g., robot task events are internal to the system and thus known with perfect accuracy). However, of the events that could suffer from duplicate reporting, SAGE eliminated numerous duplicate events. Thus, overall, SAGE reduced operator distraction significantly.

TABLE III. EVENTS ELIMINATED BY DUPLICATE MERGING

	Phase 1	Phase 2	Phase 3
Mergeable Events Created	106	14	36
Duplicates Merged	33	2	11
%	31.13	14.28	30.56

SAGE’s event priority and display system is designed to give preferential treatment to higher priority events, so that operators do not miss important information just because some lower priority information was reported first (see section III-D). Table IV shows the number of events that were created at each priority that were actually displayed in SAGE. Across all phases, operators were notified of all high priority events, and about half of medium and low priority events. The exception was Phase 2, where many more medium and low priority events were reported (but without impacting the high priority events). This is due to the distribution of the events during the phase; SAGE only fails to report an event if there are no side panels available due to concurrent events of the same or higher priority. Overall, this shows that SAGE successfully prioritized events for the operators, ensuring that the most important events were displayed.

Finally, Team Michigan had the highest robot-to-user ratio: 2 operators to 14 robots, or 7-to-1. The SAGE display was explicitly designed to make it easy to maintain SA even under such high load. Additionally, as SAGE is completely automatic, there was zero interaction time necessary to maintain the display. Figure 11 shows the amount of time the task operator interacted with any display across all three phases of the MAGIC competition. Importantly, as a zero-interaction display, SAGE is not directly responsible for any part of this interaction time. Of course, there is no way to know what indirect

impact SAGE may have had (e.g., prompting the task operator to take some action he otherwise would not have, or eliminating the need for some action he otherwise would have taken).

We encourage readers to view the video of the SAGE interface in action in the MAGIC competition [12].

TABLE IV. EVENTS DISPLAYED TO OPERATORS

	Event Priority		
	High	Medium	Low
Phase 1			
Events Created (post merge)	48	62	73
Events Displayed	48	35	39
%	100.00	56.45	53.42
Phase 2			
Events Created (post merge)	68	5	12
Events Displayed	68	5	11
%	100.00	100.00	91.67
Phase 3			
Events Created (post merge)	12	11	25
Events Displayed	12	5	16
%	100.00	45.45	64.00

V. CONCLUSIONS AND FUTURE WORK

SAGE advances the state of the art in situation awareness for teams of robots by aiding the operator’s perception of important information, comprehension of that information in context, and projection of the immediate future. It accomplishes this via a number of features, including dynamic view control, event detection and prioritization, uncertainty display, view-invariant markers, and animations. The SAGE display is completely automated, and thus does not place interaction constraints on the user.

There are other aspects of the problem of robot command and control that can complement and build upon this effort, including robot tasking and robot autonomy. The MAGIC competition showed that these two capabilities go hand-in-hand for successful robot team command and control. Future efforts should look at improving tasking efficiency by providing both better OCU tasking interfaces (possibly leveraging some of the techniques described here) and more intelligence

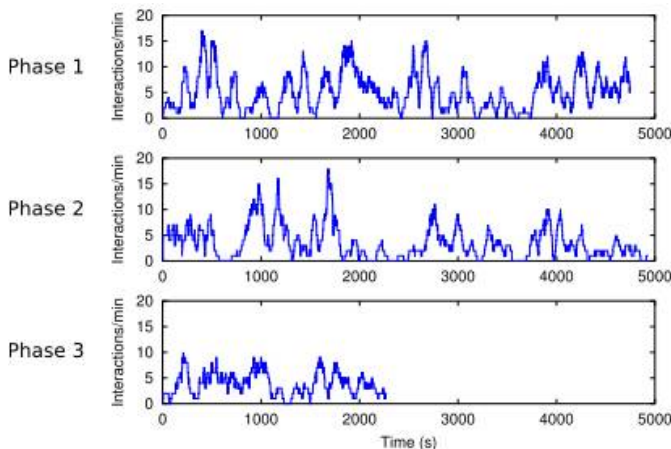


Figure 11. Task operator interactions per minute across the three phases of the MAGIC competition. Peaks indicate more operator interactions at that time. Adapted from [3].

onboard the UVs themselves. A display that integrates situation awareness and robot control is an obvious future step. Our ongoing smart interaction device (SID) research [7] will complement this work by enabling natural multi-modal interactions with the UV team, potentially obviating the need for a physical control interface at all in many situations. In future work we anticipate integrating SAGE with these and other similar technologies developed by third parties.

There is also an opportunity to use other efforts and future work to enhance the core capabilities of SAGE. In particular, SAGE’s fusion algorithms are very simple, and it lacks any tracking. Our team, along with a broad range of other experts, is developing many more sophisticated algorithms for things such as object identification, behavior detection, intent recognition, and dangerous object detection. These algorithms can be integrated within SAGE where they can provide a higher level of SA to the operator and even to the UVs themselves. Thus, future work incorporating advanced fusion methods is expected.

REFERENCES

- [1] MAGIC website: <http://www.dsto.defence.gov.au/MAGIC2010/>
- [2] P. Ranganathan, R. Morton, A. Richardson, J. Strom, R. Goedel, M. Bulic, and E. Olson. *Coordinating a Team of Robots for Urban Reconnaissance*. In Proc. of the Land Warfare Conference (LWC '10, Brisbane), 2010.
- [3] E. Olson, J. Strom, R. Morton, A. Richardson, P. Ranganathan, R. Goedel, M. Bulic, J. Crossman, B. Marinier. Progress towards multi-robot reconnaissance and the MAGIC 2010 competition. *Journal of Field Robotics*, in press.
- [4] H.A. Yanco and J. Drury, “Where am I?” *Acquiring situation awareness using a remote robot platform*, in *IEEE International Conference on Systems, Man and Cybernetics*, 2004, p. 2835-2840.
- [5] Talon OCU photograph obtained from the web at: <http://www.clemson.edu/ces/crb/students/vilas/photographs/swri05.htm>
- [6] B. Yamauchi, “PackBot: A Versatile Platform for Military Robotics”, In Proceedings of SPIE Vol. 5422: Unmanned Ground Vehicle Technology VI, Orlando, FL, April 2004.
- [7] G. Taylor, R. Frederiksen, J. Crossman, M. Quist, P. Theisen. “A Multi-Modal Intelligent User Interface for Supervisory Control of Unmanned Platforms.” (to appear) in CTS Collaborative Robots and Human Robot Interaction Workshop. Denver, CO, in press.
- [8] D.N. Powell, G. Gilbreath, M.H. Bruch, “Multi-robot operator control unit”, SPIE Proc. 6230: Unmanned Systems Technology VIII, Defense & Security Symposium, Orlando, FL, April 2006.
- [9] B. Laroche, G. Kruijff, N. Smets, T. Mioch, P. Groenewegen. “Establishing Human Situation Awareness Using a Multi-Modal Operator Control Unit In An Urban Search & Rescue Human-Robot Team,” 20th IEEE International Symposium on Robot and Human Interactive Communication, Atlanta, Georgia, 2011.
- [10] M. Endsley. “Toward a theory of situation awareness in dynamic systems,” *Human Factors*, vol. 37, no. 1, 1995, pp. 32–64.
- [11] J. Riley, L. Strater, S. Chappell, E. Connors, and M. Endsley, “Situation awareness in human-robot interaction: Challenges and user interface requirements,” in *Human-Robot Interactions in Future Military Operations*, ser. Human Factors in Defence, M. Barnes and F. Jentsch, Eds. Ashgate, 2010, pp. 171–192.
- [12] SAGE: Intelligent Robot Interface for Robot Teams (video): <https://www.youtube.com/watch?v=Cyd97VWpiek>
- [13] A. Huang, E. Olson, D. Moore. *LCM: Lightweight Communications and Marshalling*. Proceedings of the (IEEE/RSJ) International Conference on Intelligent Robotics and Systems (IROS), 2010.