CLSTR: Capability-Level System for Tracking Robots

Alexandra Bejarano[1], Claire Bonial[2], and Tom Williams[3]

Abstract— **For human operators to effectively task teams of robots, it is critical that they maintain situational awareness about the status of those robots. However, maintaining this situational awareness becomes particularly difficult when there are dynamic changes not only in the members of the robot team, but also in the capabilities of those robots. Prior work has shown that situational awareness can be supported through interfaces that effectively visualize task-relevant information. As such, in this work, we introduce a Capability-Level System for Tracking Robots (CLSTR), a new visualization for supporting operators to maintain an appropriate level of situational awareness over the capabilities of a dynamic robot team. In evaluating CLSTR through an online human-subject study (n=123), we found that a combination of different visual elements within an interface like the use of icons to summarize robot capabilities and animations to indicate team changes can help operators maintain awareness over robot teams.**

I. Introduction and Related Work

For human operators to effectively task teams of robots, it is critical that they maintain situational awareness about the status of those robots [1], [2]. This status information may include details about each robot's abilities and condition, the environment in which a robot is situated, and the tasks to be completed by that robot. To support operators in maintaining an appropriate level of situational awareness, it is essential for user interfaces to present this type of information in a way that is "readily available, easily interpretable, appropriately prominent, and simple enough for the typical user" [3]. Failure to satisfy these guidelines can result in low situational awareness, limiting an operator's ability to effectively assess and task robots, in turn negatively impacting task performance and completion [4], [5], [6]. As such, it is vital to design user interfaces that help operators maintain an appropriate level of situational awareness during robot tasking [7], [8], [9].

Previous research has shown how operators' situational awareness can be increased by designing interfaces that more effectively visualize task-relevant information through robot status indicators and interpreted sensor readings [10], [11], [12], [5]. For instance, Larochelle and Kruijff [10] designed an interface that presents operators with multiple customizable views of sensor data visualizations. While multiple views like those presented by Larochelle and Kruijff can increase an operator's situational awareness during single robot tasking, such designs may result in low situational awareness for operators tasking multiple robots because of the increased number of sensor-producing architectural components to monitor across a robot team [13], [14], [15].

In multi-robot tasking, multiple operators, each overseeing a single robot on separate user interfaces, may be needed to maintain an appropriate level of situational awareness over a team of robots. However, operators may prefer to be able to oversee multiple robots at a time to increase task efficiency [16] and reduce both the number of operators and separate interfaces needed [2]. In such cases, often a single operator has to actively switch their attention between multiple interface views to focus on different robots (e.g. as done in [17], [18], [8]). Although this enables a single operator to task multiple robots, it may be difficult for operators to maintain awareness of when and what changes or failures occur across an entire team of robots.

To address this challenge, recent work has introduced novel visualizations that quickly communicate information about the condition of multiple robots. For example, Seo et al. [19] use icons to communicate information like robot battery life and the amount of physical damage taken by a robot. Similarly, Petlowany et al. [20] introduced an augmented reality interface that visualizes pop-up menus containing robot information when a user focuses their gaze on a particular robot. However, these visualizations may still require operators to shift their attention between robots. Moreover, the information provided does not summarize the capabilities of each individual robot in a team. Without such a summary, it may be difficult for operators to quickly identify which robots are available for tasking and what those available robots can be tasked to do. This raises the research question: *How can we support operators in maintaining an appropriate level of situational awareness over a robot team?*

To address this question, we introduce a new visualization called Capability-Level System for Tracking Robots (CLSTR). In the following sections, we discuss our designed CLSTR visualization and present the results of an online human-subject study (n=123) evaluating it.

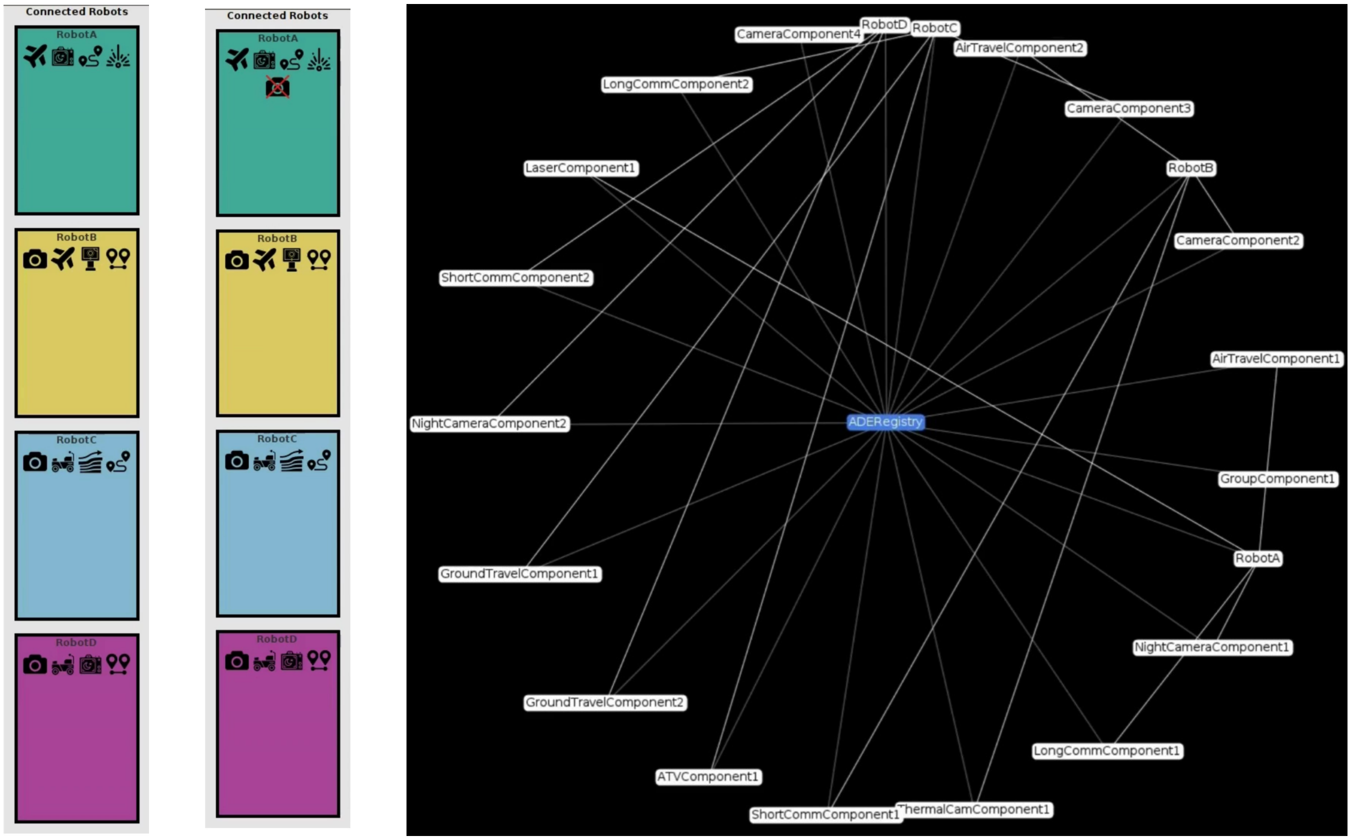
II. Capability-Level System for Tracking Robots

CLSTR is a new visualization for tasking interfaces that displays a summary of the capabilities of each individual robot in a team. Through this visualization, we aimed to support operators in maintaining an appropriate level of situational awareness over a robot team during multi-robot tasking. In this section, we reflect on the robot architecture used to enable the visualization of CLSTR, and discuss the design of this visualization.

[1]Alexandra Bejarano is a PhD student at Colorado School of Mines, Colorado, USA abejarano@mines.edu

[2]Claire Bonial is a research scientist at DEVCOM Army Research Laboratory, Maryland, USA claire.n.bonial.civ@army.mil

[2]Tom Williams is associate professor of computer science at Colorado School of Mines, Colorado, USA twilliams@mines.edu



(a) CLSTR1 (b) CLSTR2 (c) DIARC's Component Graph

Fig. 1: Interface visualizations for tracking robot components. Figures (a) and (b) show variations of CLSTR, a new visualization that makes use of icon representations to provide an organized summary of robot components. Figure (c) shows the default component graph of the DIARC robot architecture. Unlike Figures (a) and (c), (b) explicitly indicates component disconnections by showing crossed out icons.

A. Architectural Reflection

To implement our CLSTR visualization (two versions of which are shown in Figures 1a and 1b), we leverage the DIARC (Distributed Integrated Affect Reflection and Cognition) robot architecture [21], [22]. In DIARC, each robot is comprised of a set of distributed architectural components. These components include a robot's different sensing, movement, and communication abilities. As such, to track each robot and their capabilities in our CLSTR visualization, we created a grouping architectural component (GroupComponent) which performs reflection into the architecture's state in order to track what extant components are associated with which robots in a team.

To dynamically track when robot components (c) connect and disconnect from the system, the GroupComponent interacts with DIARC's central registry node to receive notifications of those connections/disconnections. When a component connects, that component is initialized with a group identifier (e.g. RobotA or RobotB) that indicates what physical robot it is a part of. Given a set of distributed components (C), the GroupComponent uses these group identifiers to partition C into r component groups $G_0, ..., G_r$ where $G = c_0, ..., c_n$. Each component group represents the set of n components that comprise a single physical robot. Using this information, we are then able to summarize the capabilities provided by the components associated with each robot, and visualize that information through CLSTR.

B. Design

In this section, we discuss the design of multiple versions of the CLSTR robot architecture visualization. To help operators quickly identify which robots are available for tasking and what those available robots can be tasked to do, we aimed to provide operators with a visual summary of the members of a robot team, and the capabilities of each of those robots. Based on design recommendations by Oury and Ritter [3], CLSTR was designed to present this summary in such a way that is "readily available, easily interpretable, appropriately prominent, and simple enough for the typical user".

To be readily available and appropriately prominent, CLSTR was designed as as compact side-panel for an existing tasking interface, thus providing a persistent summary of the robot team that updates whenever team changes occur (e.g., robot components connect and disconnect). With such a summary, operators can avoid having to actively switch their attention between multiple interface views to focus on different robots.

To be easily interpretable and simple enough for the typical user, CLSTR was designed to use contrasting colors and representative icons. Contrasting colors were used to help operators easily differentiate between different component groups associated with different physical robots, and icons were used to represent different robot capabilities to help users easily identify each robot's current capabilities [23]. For instance, an icon of a camera silhouette can be used to indicate that a robot has a vision-based sensing ability. This use of icons is similar to Seo et al. [19]'s use of emojis in which simple representations are used to reduce the cognitive processing needed to interpret information about robots. Overall, these design choices were used to summarize robot team information in a simple and organized way to support operators in maintaining an appropriate level of situational awareness over a robot team.

Two versions of our CLSTR visualization are shown in Figures 1a and 1b, each of which shows information about four robots (Robot A, B, C, D). These visualizations are updated anytime robot components connect and disconnect. When a component connects, a new icon is added under the relevant component group. As such, Figures 1a and 1b both show that each robot has four different available capabilities, as shown by the different component group colors and set of icons. When a component disconnects, this is visualized in one of two ways, as we will now describe.

Figure 1a shows our first version of CLSTR (CLSTR1), which visualizes disconnections using a similar design pattern as DIARC's default architectural visualization. As shown in Figure 1c, DIARC's default visualization is a component graph that represents the architecture's state as a set of nodes (one for each architectural component), with lines connecting those nodes representing connections between components. In this visualization, when a component disconnects, that component is removed entirely from the graph. Similarly, in our CLSTR1 visualization, when a component disconnects, the associated icon is removed. However, this design only shows to operators the available robot capabilities, possibly making it difficult for operators to realize and remember that something changed among the robot team. As such, we designed a second version of CLSTR (CLSTR2) in which, when a component disconnects, the associated icon is instead crossed out and labeled as unavailable to indicate that a particular capability is no longer available. This second design was intended to clearly present information about both currently available and no-longer-available robot capabilities.

III. Evaluation

To evaluate CLSTR, we conducted an IRB-approved online human-subjects study on the Prolific survey platform, in which participants watched videos of our CLSTR visualization and DIARC's component graph while completing a secondary task.

This experiment was designed to test two key hypotheses:

- **H1:** CLSTR will enable operators to more accurately recognize changes to a robot team than DIARC's component graph.
- **H2:** CLSTR2 will enable operators to more accurately recognize component disconnections than CLSTR1.

A. Experimental Design and Procedure

This study followed a mixed factorial design with visualization type (CLSTR vs DIARC's component graph) and scenario as within subjects, and CLSTR version (CLSTR1 vs CLSTR2) as a between-subjects variable. As such, each participant was shown videos of DIARC's component graph and of either CLSTR1 or CLSTR2. Videos were counterbalanced to vary the order in which they were shown.

After providing informed consent and demographic information, participants performed six rounds of an article counting task in which participants counted the number of instances of the letter 'a' within a text paragraph. This task has been used in previous Human-Robot Interaction experiments to split participant's attention away from a monitored robot [24], [25]. As such, in this work we used this same task to split participants' attention away from videos of either CLSTR or DIARC's component graph. In these visualization videos, robot components were shown connecting and disconnecting based on the scenarios detailed in Section III-B. Of the six rounds watched by participants, three rounds included a video of CLSTR, and three rounds included a video of DIARC's component graph.

In each round, participants performed an article counting task while a video of CLSTR or DIARC's component graph was shown in a side panel. Participants were instructed to passively attend to changes within the video while completing their counting task. Each video was about 1 minute in length and was unable to be replayed. When the video finished playing, participants were asked to indicate the number of instances they had counted to ensure they had been performing the counting task. Participants were then asked awareness questions about the visualization video shown. These questions are detailed in Section III-C.

Once all rounds were completed, participants were asked a final free response question: "Of the two [visualizations] shown throughout this evaluation, which would you prefer to track the components of multiple robots? Please write at least 1-2 sentences to explain your answer."

B. Scenario Design

For each of the two visualizations (CLSTR and DIARC's component graph), participants performed three rounds of the task described above. During each of these rounds, a different scenario involving a team of four robots was visualized in the visualization panel. These three scenarios had increasing levels of complexity.

1) Scenario 0: In the first round for each visualization, the visualization video depicted a simple scenario in which one new robot component connects and a different component disconnects. This round was used as a practice round to introduce participants to the relevant visualization. As such, participant responses to the questions in these practice rounds were not analyzed.

2) Scenario 1: Scenario 1 demonstrated one new robot component connecting, and an entire robot disconnecting (i.e. all components of the same component group disconnected), with the remaining two robots' capabilities not changing during the scenario.

3) Scenario 2: Scenario 2 demonstrated two new robot components connecting and two robot components disconnecting (following an alternating pattern between component connections and disconnections), leaving the capabilities of only one robot unchanged.

C. Measures

To assess the participant's situational awareness during each of these three scenarios, we presented participants with the following awareness questions at the end of each round:

1) "During the video, how many components seem to have disconnected from the interface?"
2) "During the video, how many new components seem to have connected to the interface?"
3) "Which robots remained the same throughout the entire video?"

Finally, participants were asked: "What difficulties did you face with the interface during the round?"

D. Participants

123 participants were recruited from Prolific (54 male, 68 female, 1 non-binary). Participants ranged from 18 to 74 years old (M=35.358, SD=11.547). 62 participants were shown videos of CLSTR1, 61 participants were shown videos of CLSTR2, and all participants were shown videos of DIARC.

E. Analysis

For each awareness question, accuracy was calculated using the following formula:

$$100 - \left(\frac{|ObservedValue - ActualValue|}{ActualValue} * 100 \right)$$

A Bayesian statistical analysis was then conducted on anonymized data using the JASP statistical software [26]. This analysis was comprised of a set of Repeated-Measures (RM) ANOVAs with Bayes Factor (BF) Analyses. Specifically, Inclusion BFs across Matched Models [27], [28] were calculated through Bayesian Model Averaging. The Inclusion BFs produced by this approach represent the strength of evidence in favor of models including each candidate main effect or interaction effect. All BFs reported for RM-ANOVAs are thus $BF_{Incl_{10}}$, i.e., Inclusion BFs representing the odds ratio of evidence in favor of an effect (H_1) versus evidence against an effect (H_0). To discuss these results, we use the linguistic interpretations of reported BFs as recommended by JASP reporting guidelines [29]. For all analyses, when evidence for an effect could not be ruled out (BF>0.333), the results were further analyzed using post-hoc Bayesian t-tests.

To test our hypotheses, each measure was analyzed with two Bayesian RM-ANOVAs. To compare CLSTR and DIARC's component graph, the first RM-ANOVA included visualization type and scenario as repeated measures factors. To compare CLSTR1 and CLSTR2, the second RM-ANOVA included scenario as a repeated measures factor and visualization type as a between subjects factor.

All study videos, data, and analysis scripts are available via the Open Science Framework at `https://osf.io/hbr2n/`.

IV. Results

In this section, we describe the results of the analyses described above.

A. CLSTR vs DIARC's Component Graph

1) Component Disconnects: Moderate evidence was found in favor of an effect of visualization on disconnect accuracy (BF=3.858). This result indicates that participants more accurately indicated the number of component disconnects with CLSTR (M=60.501, SD=35.321) than with DIARC's component graph (M=50.678, SD=28.761) regardless of scenario. This provides evidence for H1.

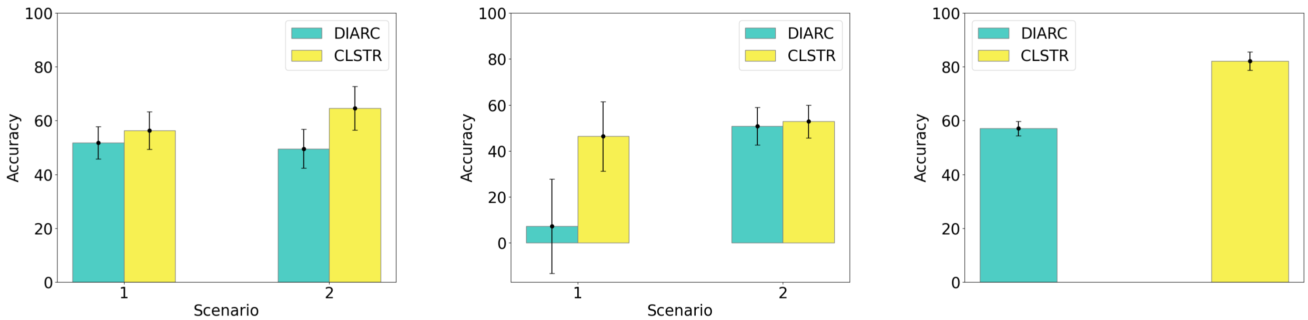
Moderate evidence was found against an effect of scenario on disconnect accuracy (BF=0.179). This result suggests that scenario alone did not influence how accurately participants indicated the number of component disconnects.

Anecdotal evidence was found against an interaction effect between visualization and scenario disconnect accuracy (BF=0.814). Post-hoc t-tests indicated moderate evidence against a difference in disconnect accuracy between CLSTR and DIARC's component graph in Scenario 1 (BF=0.286), but strong evidence in favor of such a difference in Scenario 2 (BF=23.544). As shown in Figure 2a, these results indicate that in Scenario 1 participants performed similarly with both visualizations, while in Scenario 2 participants more accurately indicated the number of component disconnects with CLSTR (M=64.634, SD=45.598) than with DIARC's component graph (M=49.593, SD=40.739).

2) Component Connects: Strong evidence was found in favor of an effect of visualization on connect accuracy (BF=12.269). This result indicates that participants more accurately indicated the number of component connects with CLSTR (M=49.593, SD=51.413) than with DIARC's component graph (M=29.065, SD=64.606) regardless of scenario. This provides evidence for H1.

Very strong evidence was found in favor of an effect of scenario on connect accuracy (BF=32.522). This result indicates that participants more accurately indicated the number of component connects in Scenario 2 (M=51.829, SD=33.750) than in Scenario 1 (M=26.829, SD=80.798) regardless of visualization.

Strong evidence was found in favor of an interaction effect between visualization and scenario on connect accuracy (BF=29.100). As shown in Figure 2b, these results indicate that in Scenario 1 participants more accurately indicated the number of component connects with CLSTR (M=



(a) Component Disconnects (b) Component Connects (c) Robots that Remained the Same

Fig. 2: Accuracy comparison between DIARC's component graph and CLSTR. In all figures, error bars represent 95% credible interval.

46.341, SD= 84.245) than with DIARC's component graph (M=7.317, SD=115.355), while in Scenario 2 participants performed similarly with both visualizations.

3) Robots that Remained the Same: Extreme evidence was found in favor of an effect of visualization on participant ability to identify which robots had not changed during each round (BF=2.382×10^{20}). As shown in Figure 2c, these results indicate that participants more accurately indicated which robots remained the same with CLSTR (M=82.114, SD=19.010) than with DIARC's component graph (M=57.114, SD=15.274) regardless of scenario. This provides evidence for H1.

Moderate evidence was found against an effect of scenario (BF=0.190) and against an interaction effect between visualization and scenario (BF=0.150). These results suggest that scenario did not influence how accurately participants indicated which robots remained the same.

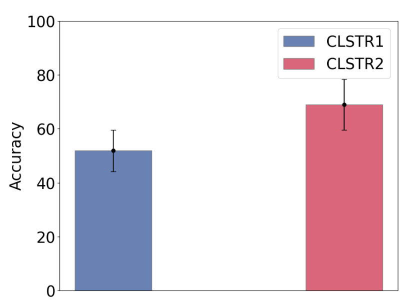


Fig. 3: Disconnect accuracy comparison between CLSTR versions.

B. CLSTR1 vs CLSTR2

1) Component Disconnects: Moderate evidence was found in favor of an effect of CLSTR version on disconnect accuracy (BF=5.632). As shown in Figure 3, these results indicate that participants more accurately indicated the number of component disconnects with CLSTR2 (M=68.952, SD=37.728) than with CLSTR1 (M= 51.913, SD=30.671) regardless of scenario. This provides evidence for H2.

Anecdotal evidence was found against an effect of scenario on disconnect accuracy (BF=0.880). This result indicates that there may be an effect of scenario, but more data would be needed to support or rule out this effect.

Moderate evidence was found against an interaction effect between CLSTR version and scenario on disconnect accuracy (BF=0.174).

2) Component Connects: Anecdotal evidence was found against an effect of CLSTR version on connect accuracy (BF=0.386). This result suggests that there may be an effect of CLSTR version, but more data would be needed to support or rule out this effect.

Moderate evidence was found against an effect of scenario on connect accuracy (BF=0.198). This result suggests that scenario alone did not influence how accurately participants indicated the number of component connects.

Anecdotal evidence was found in favor of an interaction effect between CLSTR version and scenario on connect accuracy (BF=1.288). Post-hoc t-tests indicated anecdotal evidence against a difference in connect accuracy in Scenario 1 between CLSTR versions (BF=0.807), and moderate evidence against a difference in connect accuracy in Scenario 2 between CLSTR versions (BF=0.203). These results suggest that there may be an interaction effect between CLSTR version and scenario on connect accuracy, but more data would be needed to support or rule out this effect.

3) Robots that Remained the Same: Moderate evidence was found against an effect of CLSTR version on participant ability to identify which robots had not changed during each round (BF=0.287). Moderate evidence was found against an effect of scenario (BF=0.168). Moderate evidence was found against an interaction effect between CLSTR version and scenario (BF=0.231). Overall, these results indicate that when it comes to indicating which robots remained the same, participants performed similarly regardless of CLSTR version or scenario.

C. Participant Feedback

Overall, participants indicated a greater preference for CLSTR than DIARC's component graph for tracking the components of multiple robots (100 participants preferred CLSTR, 18 preferred DIARC's component graph, 5 preferred both for different instances). Participants indicated that with CLSTR, it was easier to process and distinguish between component groups because of the visual grouping and colors. CLSTR was also described as being easier for deciphering *what* changes happened in the different scenarios. Whereas, with DIARC's component graph, participants indicated having difficulty identifying whether a component connected or disconnected as it was hard to track the moving graph lines and read component labels.

However, some participants found DIARC's component graph to be more helpful in noticing *when* a change happened. For instance, one participant commented "The movement of [DIARC] was distinguishable even in my peripheral field of vision. I felt like it made it easier to know that a change was occurring. As far as noticing overall changes after the video, I think [CLSTR] is more efficient at being able to identify if components were removed."

V. Discussion

Overall, our results clearly supported hypotheses 1 and 2.

Our results indicate that CLSTR can more effectively provide information about changes within a robot team. In particular, we found that our organized summary of robot capabilities through icon representations made it easier for participants to determine *what* changes occurred (number of disconnects and connects) among a robot team as well as what robots remained unchanged. This suggests that this type of visualization of robot capabilities can improve situational awareness for operators through its presentation of simple and easily interpretable information about a robot team. Moreover, we found that "crossing out" disconnected components in CLSTR2 can more effectively ensure that participants notice and understand that a change occurred than with CLSTR1's visualization.

While CLSTR more effectively provided robot team information than DIARC's component graph, some participants noted how the use of motion within a visualization (such as changing graph segments) can be helpful in realizing *when* a change in a robot team occurs. This motion was indicated as being particularly important when participants were engaged in a secondary task, as the motion helped redirect their attention to changes occurring within the robot team. This suggests that including dynamic elements into visualizations could be beneficial in monitoring a robot team. Similarly, other cues like sound can also be paired with visualizations to help shift an operator's attention to particular changes among a robot team [30].

Although the use of visualizations like icons and motion can help support situational awareness over a robot team, interface designers must balance the use of these design cues to avoid overloading an operator's visual processing. For instance, interface designers may want to avoid the use of too many icons as this may simply increase the number of elements operators need to keep track of and understand. Similarly, interface designers may also want to avoid the inclusion of too many animations that may distract from important elements or increase the difficulty of interpreting visualizations.

Overall, our findings suggest that effective methods for tracking the availability of a dynamic robot team should include visual strategies that enhance both the detection of robot team changes and the timing of those changes.

A. Limitations and Future Work

Although our participants viewed scenarios based on real-world multi-robot systems, participants were not familiarized with the details of the robot teams depicted in those visualizations. As such, it may be valuable in future work to debrief participants on what robot component/capability is represented by which icons. If participants were debriefed in this way, this would provide an opportunity to better assess participants' recognition of *which* components have changed. And, since the appearance of the icons may need to vary across different task contexts, these future participants could provide feedback on which type of icons best represent different robot capabilities in specific contexts.

Moreover, to evaluate CLSTR in a way that provides increased ecological validity, future work may examine CLSTR's performance in contexts where participants are actively taking part in robot tasking rather than the highly controlled article counting task used in this work.

VI. Conclusion

In this work, we introduce CLSTR, a new visualization for supporting operators to maintain an appropriate level of situational awareness over the capabilities of a dynamic robot team through simple icon visualizations. Through an online human-subject study evaluating our design, we found that CLSTR is able to better maintain operators' situational awareness of the changes occurring within a robot team, especially when using CLSTR2's method of visualizing recently disconnected components. Moreover, our results demonstrate aspects of older visualizations (e.g., salient motions) that may further enhance operators' situational awareness. Overall, our results provide clear guidance for how robot architecture designers can develop interfaces that allow users of those architectures to maintain high situational awareness about the state of the robots comprising those architectures.

ACKNOWLEDGMENT

This work was supported in part by ARL DCIST grant W911NF-17-2-0181.

References

[1] S. Opiyo, J. Zhou, E. Mwangi, W. Kai, and I. Sunusi, "A review on teleoperation of mobile ground robots: Architecture and situation awareness," *International Journal of Control, Automation and Systems*, vol. 19, pp. 1384–1407, 2021.

[2] A. Bejarano, S. Elbeleidy, T. Mott, S. Negrete-Alamillo, L. A. Armenta, and T. Williams, “Hardships in the land of oz: Robot control challenges faced by hri researchers and real-world teleoperators,” 2024.
[3] J. D. Oury and F. E. Ritter, “How user-centered design supports situation awareness for complex interfaces,” *Building Better Interfaces for Remote Autonomous Systems: An Introduction for Systems Engineers*, pp. 21–35, 2021.
[4] M. R. Endsley, “Toward a theory of situation awareness in dynamic systems,” *Human factors*, vol. 37, no. 1, pp. 32–64, 1995.
[5] D. J. Rea and S. H. Seo, “Still not solved: A call for renewed focus on user-centered teleoperation interfaces,” *Frontiers in Robotics and AI*, vol. 9, p. 704225, 2022.
[6] E. K. Chiou, M. Demir, V. Buchanan, C. C. Corral, M. R. Endsley, G. J. Lematta, N. J. Cooke, and N. J. McNeese, “Towards human–robot teaming: Tradeoffs of explanation-based communication strategies in a virtual search and rescue task,” *International Journal of Social Robotics*, pp. 1–20, 2022.
[7] A. Rule and J. Forlizzi, “Designing interfaces for multi-user, multi-robot systems,” in *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, 2012, pp. 97–104.
[8] J. J. Roldán, E. Peña-Tapia, A. Martín-Barrio, M. A. Olivares-Méndez, J. Del Cerro, and A. Barrientos, “Multi-robot interfaces and operator situational awareness: Study of the impact of immersion and prediction,” *Sensors*, vol. 17, no. 8, p. 1720, 2017.
[9] S. Chen, M. J. O’Brien, F. Talbot, J. Williams, B. Tidd, A. Pitt, and R. C. Arkin, “Multi-modal user interface for multi-robot control in underground environments,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 9995–10 002.
[10] B. Larochelle and G.-J. M. Kruijff, “Multi-view operator control unit to improve situation awareness in usar missions,” in *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, 2012, pp. 1103–1108.
[11] M. Moniruzzaman, A. Rassau, D. Chai, and S. M. S. Islam, “Teleoperation methods and enhancement techniques for mobile robots: A comprehensive survey,” *Robotics and Autonomous Systems*, vol. 150, p. 103973, 2022.
[12] A. Ramesh, M. Englund, A. Theodorou, R. Stolkin, and M. Chiou, “Robot health indicator: A visual cue to improve level of autonomy switching systems,” *arXiv preprint arXiv:2303.06776*, 2023.
[13] C. Y. Wong and G. Seet, “Workload, awareness and automation in multiple-robot supervision,” *International Journal of Advanced Robotic Systems*, vol. 14, no. 3, p. 1729881417710463, 2017.
[14] W. Jo, R. Wang, B. Yang, D. Foti, M. Rastgaar, and B.-C. Min, “Affective workload allocation for multi-human multi-robot teams,” *arXiv preprint arXiv:2303.10465*, 2023.
[15] A. Dahiya, A. M. Aroyo, K. Dautenhahn, and S. L. Smith, “A survey of multi-agent human–robot interaction systems,” *Robotics and Autonomous Systems*, 2023.
[16] Y. Jia, N. Xi, and J. Buether, “Design of single-operator-multi-robot teleoperation systems with random communication delay,” in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 171–176.
[17] J. Y. Chen and M. J. Barnes, “Supervisory control of multiple robots: Effects of imperfect automation and individual differences,” *Human Factors*, vol. 54, no. 2, pp. 157–174, 2012.
[18] F. Niroui, Y. Liu, R. Bichay, E. Barker, C. Elchami, O. Gillett, M. Ficocelli, and G. Nejat, “A graphical user interface for multi-robot control in urban search and rescue applications,” in *2016 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)*. IEEE, 2016, pp. 217–222.
[19] S. H. Seo, J. E. Young, and P. Irani, “How are your robot friends doing? a design exploration of graphical techniques supporting awareness of robot team members in teleoperation,” *International Journal of Social Robotics*, vol. 13, no. 4, pp. 725–749, 2021.
[20] C. Petlowany and N. Hahn, “Gaze-based augmented reality interfaces to support human-robot teams,” 2024.
[21] M. Scheutz, G. Briggs, R. Cantrell, E. Krause, T. Williams, and R. Veale, “Novel mechanisms for natural human-robot interactions in the diarc architecture,” in *Proceedings of AAAI Workshop on Intelligent Robotic Systems*. Palo Alto, CA, 2013, p. 66.
[22] P. W. Schermerhorn, J. F. Kramer, C. Middendorff, and M. Scheutz, “Diarc: A testbed for natural human-robot interaction.” in *AAAI*, vol. 6, 2006, pp. 1972–1973.
[23] A. Blair-Early and M. Zender, “User interface design principles for interaction design,” *Design Issues*, vol. 24, no. 3, pp. 85–107, 2008.
[24] R. Wen, B. Kim, E. Phillips, Q. Zhu, and T. Williams, “On further reflection... moral reflections enhance robotic moral persuasive capability,” in *International Conference on Persuasive Technology*. Springer, 2023, pp. 290–304.
[25] ——, “Comparing norm-based and role-based strategies for robot communication of role-grounded moral norms,” *ACM Transactions on Human-Robot Interaction*, vol. 12, no. 3, pp. 1–25, 2023.
[26] JASP Team, “JASP (Version 0.18.3)[Computer software],” 2024. [Online]. Available: https://jasp-stats.org/
[27] M. Hinne, Q. F. Gronau, D. van den Bergh, and E.-J. Wagenmakers, “A conceptual introduction to bayesian model averaging,” *Advances in Methods and Practices in Psychological Science*, 2020.
[28] S. Mathôt, “Bayes like a baws: Interpreting bayesian repeated measures in jasp,” *Cog. Sci. and more. Retrieved from: https://www. cogsci. nl/blog/interpreting-bayesian-repeated-measures-in-jasp*, 2017.
[29] J. Van Doorn, D. Van Den Bergh, U. Böhm, F. Dablander, K. Derks, T. Draws, A. Etz, N. J. Evans, Q. F. Gronau, J. M. Haaf *et al.*, “The jasp guidelines for conducting and reporting a bayesian analysis,” *Psychonomic Bulletin & Review*, vol. 28, pp. 813–826, 2021.
[30] S. GRUSHKO, A. VYSOCKY, J. SUDER, L. GLOGAR, Z. BOBOVSKY *et al.*, “Improving human awareness during collaboration with robot,” *MM Science Journal*, no. 6, pp. 5475–5480, 2021.