Taming Multiple Chat Room Collaboration: Real-Time Visual Cues to Social Networks and Emerging Threads

Lindsley G. Boiney, PhD
The MITRE Corp
lboiney@mitre.org

Bradley Goodman, PhD
The MITRE Corp
bgoodman@mitre.org

Robert Gaimari , Jeffrey Zarrella, Christopher Berube & Janet Hitzeman, PhD
The MITRE Corp
rgaimari@mitre.org, jzarrella@mitre.org, cdb@mitre.org, hitz@mitre.org

ABSTRACT

Distributed teams increasingly rely on collaboration environments, typically including chat, to link diverse experts for real time information sharing and decision-making. Current chat-based technologies enable easy exchange of information, but don't focus on managing those information exchanges. Important cues that guide face-to-face collaboration are either lost or missing. In some military environments, operators may juggle over a dozen chat rooms in order to collaborate on complex missions. This often leads to confusion, overload, miscommunication and delayed decisions.

Our technology supports chat management. A summary display bar reduces the number of chat rooms operators need open by providing high level situational awareness pointers, in real-time, to: a) rooms with increasing message activity levels, b) rooms in which important collaborators are participating (those in the operator's social network), and c) rooms in which operator-selected keywords are used. This ability to peripherally monitor less critical chat rooms reduces operator overload, while enhancing the ability to rapidly detect important emerging discussion threads.

Keywords

Collaboration, social networks, chat, visualization, real time decision making.

INTRODUCTION

Distributed, collaborative work environments, typically including text and voice chat, offer a valuable medium to bring diverse experts together for real time information sharing and decision making. For instance, network-centric capabilities have enabled the evolution of Time Sensitive Targeting (TST) or Dynamic Targeting (DT) cells as an important weapon in combating asymmetric threats. Distributed collaborators must reach shared situational awareness (SA) of not only the task environment, but also of team processes to succeed in their mission.

Yet field observations and controlled studies of team decision-making have demonstrated breakdowns in communication and process, revealing that existing collaborative environments don't sufficiently support the necessary team behaviors and interactions. Operators are often overloaded with information, distracted by multiple sources, and unclear where other team members are focused. This can lead to missed information and delayed decisions (Boiney, 2005).

A recent MIT report on network centric warfare identified *attention allocation* as one of the eight top issues for human supervisory control (Mitchell, Cummings, and Sheridan, 2004). The changing nature of missions, the impact of new information technologies, and the need to work in richer contexts (joint, coalition, and inter-agency) have resulted in new approaches to command and control that place unique demands on both human cognition and social processes. These demands go well beyond the need to process increased quantities of information. Attention pressures are intensifying as operators must regularly multi-task, accommodate frequent interruptions, attend to numerous information sources and modalities, and scan for unanticipated events and threats (Boiney, 2007; Gonzalez & Mark, 2004; Hudson et al, 2002). These challenges for attention allocation arise in emergency management and homeland security domains, and across all military services. They are being keenly felt in distributed operations and at the tactical edge where systems must gain operator attention without distracting them unduly from current tasking. Human attention has become a critical resource for situational awareness and decision-making.

METHODS

To appreciate the way people actually interact with each other and with technologies under stressful conditions, immersion in complex collaborative environments is invaluable. One of the authors engaged in three separate weeks of direct over-the-shoulder observation of experienced operators as they carried out tasks in an experimental Coalition/Combined Air Operations Center (CAOC). Contextual analysis of collaboration activities involved a coordinated blend of three approaches: direct field observation of operators, expert interviews, and analysis of text within "chat" logs. We have found that combining and interweaving data of these three types creates a powerful mechanism for understanding collaboration in context. Observation, chat and interviews—in essence, people's actions, words, and thoughts, respectively—help reveal the interactions between the operators, their systems, and their complex environment.

Direct field observation involved serving as unobtrusive observers alongside operators in their work environment (more specifically, at several TST-related exercises and experiments that closely resemble actual operational environments). With permission from operators, it was possible to sit or stand near enough to hear conversations, see tool usage, observe when operators phoned or got up to speak to others, and detect more subtle body language suggesting frustration, confusion, fatigue, or other mental and emotional states. The ability to look over operators' shoulders allowed documentation of which applications they used, and how. In some cases, operators even allowed listening in on audio communications in real time via dual headsets. Direct observation is valuable for monitoring the full communication spectrum, from text and audio chat to face-to-face discussions and phone calls, to physical expressions and gestures that reveal emotions or cognitive states.

Second, data from semi-structured interviews of experienced operators provided important context by revealing what an operator was thinking at the time of a particular activity. Interviews allowed solicitation of operator rationales, their judgments of the amount and quality of information provided, their success stories or perceived challenges involving information sharing and coordination, and their perspectives on social issues such as the level of team familiarity, cohesiveness, and trust. Questions (such as "What prompted you to ask for that information at that point?" "How did you know to share the information with those particular individuals?" "How did you reconcile those conflicting pieces of information?") were asked before or after an exercise to avoid disrupting activities, or during a break in the action if an operator indicated it was permissible.

Third, text chat analyses involved logging and combing through the text of instant messages sent between individuals in “chat rooms”. Chat rooms were organized along convenient topics and subgroups such as dynamic targeting, weapons, intelligence, or internal coordination. Each operator joined the subset of chat rooms deemed most useful for his individual tasks, and for his information sharing needs with others on the team. Data from chat logs have the advantages of immediacy and objectivity; messages are captured word for word as they occur in real time so there is no bias or time lag between what occurred and what is recorded. The clear documentation of senders, recipients, information content, and time stamps allows re-creation of collaborative threads across multiple participants from beginning to end. (See Boiney, 2005 for full methodological details.)

LESSONS LEARNED: OPERATIONAL NEEDS

Nearly all operators observed in the CAOC had two displays in front of their stations; one was devoted almost entirely to chat rooms. Operators had an average of six to eight chat windows open and some had as many as fourteen. See Figure 1.

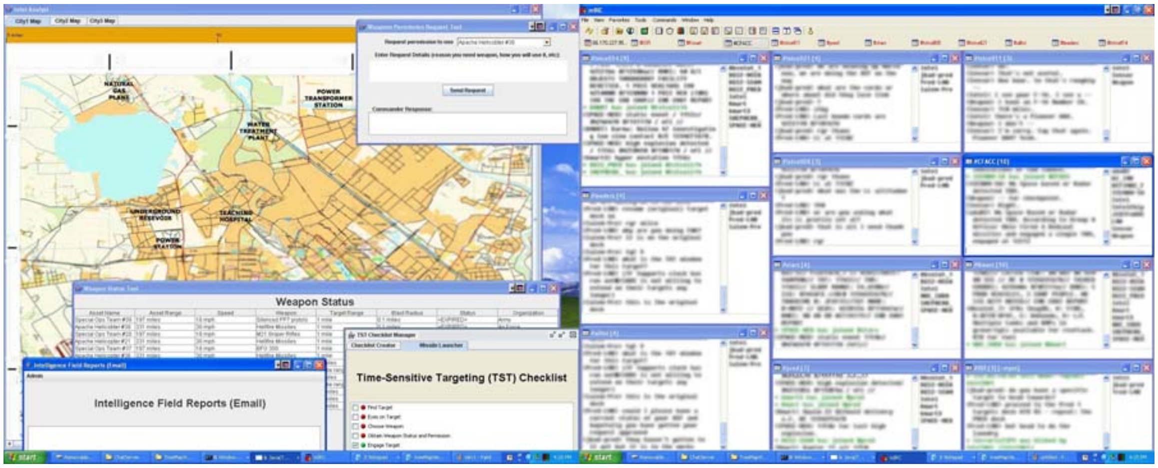

Figure 1. Maintaining collaborative SA (right) while attending to mission-specific tools (left)

Each chat room had its own unique name and set of participants, with its own purported topic. Operators had to monitor these chat windows for situational awareness (SA) while simultaneously attending to the mission-specific tools on the other display. Operators developed their own workarounds to help manage this large quantity of dynamic information. For instance, many spent time and effort cueing others to important chat events that may have been missed. Indeed, operators sometimes become so engrossed in reading chat rooms for important developments and cueing other team members that they neglected other issues. As one operator put it, “…I get tunnel vision to accomplish the coordination and lose sight of anything else.” Endsley et al (2003) define “attentional tunneling” as one of eight SA Demons:

> “When succumbing to attentional tunneling, they lock in on certain aspects or features of the environment they are trying to process, and will either intentionally or inadvertently drop their scanning behavior. In this case, their SA may be very good on the part of the environment they are concentrating on, but will quickly become outdated on the aspects they have stopped attending to.” (Endsley et. al, 2000)

Co-located operators in the CAOC would often suspend their own activities in order to verbally update a team member they noticed returning to his or her station. They might briefly summarize the activity that had occurred and indicate where among the myriad information sources to find the updates. This required not only time and effort from the cueing operator, but sufficient awareness of teammate roles and status to understand what updates would be valuable and relevant. This willingness to voluntarily keep an eye out for others’ comings and goings and cue them to high priority information helped avoid missed information and speed reacquisition of SA after a distraction or interruption. As one operator described it, "I want at least 7 rooms open—can't fit it all. I rely on the rest of the team for cues." Though none of this cooperative cueing was part of the official collaborative process or training, it evolved to become a critical part of the cognitive and social fabric of team communication.

Operators explained during interviews that they actively participated in only some of the chat rooms they had open, while they "just watched" the others. Though they often had difficulty articulating exactly what they were watching for, they were extremely reluctant to completely close those secondary windows—even if there was only enough screen real estate to maintain them as tiny, one-inch square windows. It became clear this was a *means of peripherally scanning them for cues* such as increases in room activity levels; if message traffic dramatically increased in one of these small windows, the quickly scrolling text would catch the operator's eye. If this was detected via peripheral scanning, the operator could then enlarge that window to attend more fully and read the message traffic to see if further action was needed.

Overall, we learned that operators perform several essential collaborative tasks. They:

- maintain sufficient team awareness to enable effective coordination, even when not co-located;
- cue other team members to important information, emerging events, or changing priorities;
- judge who needs—and does not need—particular information, balancing the need for sharing against the danger of cognitive overload;
- select appropriate communication modalities for sharing information of varying importance, time-sensitivity, and intended audience;
- establish trust and credibility with one another;
- validate information, and determine where to get more if needed;
- engage in collaborative sense making – handling ambiguous or conflicting information.

The following prototype design addresses the first four issues.

DESIGN CONCEPT & TECHNICAL APPROACH: "DYNAMIC CHAT MANAGER (DCM)"

The primary goal was to help operators manage collaboration via chat so as to reduce overload and distractions and improve detection of important, potentially unexpected information. We will describe development of a chat management prototype, the "Dynamic Chat Manager" (DCM), which provides real-time dynamic cues intended to help operators focus their attention on those chat rooms most likely to contain relevant information. The DCM prototype is linked to the mIRC (Internet Relay Chat) client; mIRC is very popular and its integrated scripting language makes it extensible and versatile. Our design is not dependent, however, on this particular chat client.

The first component of the design concept was to use artificial intelligence, in the form of chat "bots", to quietly monitor multiple chat rooms and discover patterns and relationships that are effortful for humans to detect on their own.

The second component of the design concept was to provide simple, intuitive, and unobtrusive visualizations of collaboration cues so operators could more easily decide in real time where their attention was most needed.

The field work described in the previous section helped us identify chat cues that, if automated, would be valuable for operator attention management and reduce their need to scan so many chat rooms. These cues can be summarized as real time indicators involving:

- ***Where***: "Which chat rooms are currently *hot* as indicated by significant message activity?"
- ***Who***: "If there's a discussion heating up, are any of my important collaborators involved in the discussion?"
- ***What***: "Are some chat rooms mentioning terms/keywords critical to my role?"

Where: Activity Level as an Indicator

"Which chat rooms are currently *hot* as indicated by significant message activity?" To address this first question, we employ chat "bots". Bots join the chat rooms to monitor conversations in the rooms, and to track increases or decreases in each chat room's message activity level. (We purposefully do not include the often numerous "join" and "leave" messages generated in a room whenever someone enters or exits; these can be highly distracting and do not reflect actual communication between operators.)

Who: **Chat Social Networks as an Indicator**

"If there's a discussion heating up, are any of my important collaborators involved in the discussion?" To address this second question, we employ social network analysis to develop models that infer, given a set of observations over time, the likelihood that someone is in a user's social network.

Our prototype employs modifications to the open source software PieSpy, a Java bot capable of monitoring a set of IRC chat channels. PieSpy uses a simple set of analytics to record relationships between pairs of users, allowing it to discover and build a mathematical model of underlying patterns and relationships reflecting which individuals interact most often with other operators. The resulting models can be used to draw the social networks and create animations of evolving social networks. Each user's social network will evolve over time and can move with an operator from session to session.

What: **Keyword Content as an Indicator**

"Are some chat rooms mentioning terms/keywords critical to my role?" To address this third question, we allow operators to specify optional keyword alerts. The prototype will highlight specified keywords within the relevant mIRC chat rooms (a capability already available through mIRC), but will additionally need to provide integrated visual cues within the high level display. We describe this feature in the next section.

VISUALIZATION

Simply providing *additional* information in real time would almost certainly be distracting to operators in this complex environment (McCrickard & Chewar, 2003; Bennett and Flach, 1992). Instead, we employ high level summary visualizations that allow operators to gain *richer contextual cues* while actually having *fewer* chat windows open. A key advantage of DCM is the ability to monitor and maintain high level situational awareness regarding multiple chat rooms (perhaps a dozen) while having the luxury to avoid cluttering your screen with anything but your few primary chat rooms.

Figure 2 shows the overall design concept, in which a slim subdivided bar at the top cues operators via size and color to these collaboration dynamics. Note that in this situation, the operator is monitoring nine rooms via the Dynamic Chat Manager display bar, while having only four critical chat rooms fully open.

DCM signals the presence of an operator's ***keyword*** by temporarily ***coloring the name*** of the corresponding chat room on the display bar. This is significant because it alerts the operator to keywords appearing within ***any*** of the rooms being monitored via DCM, as opposed to just those chat rooms fully opened on the desktop. To help operator's quickly locate the particular keyword within the discussion thread, DCM also highlights it within the relevant chat room, much as mIRC alone can do. But DCM uses more judicious highlighting—coloring only the keyword itself, rather than the entire surrounding line of chat—making it easier to glance at the room and see which keyword was detected.

As a result, DCM cues operators to the three real time variables identified to help operators decide when to pay attention to specific chat rooms: the ***who*** (social network participation shown via bar fill color), the ***where*** (high activity rooms shown via bar size), and the ***what*** (keyword use shown via brief color change of bar's text).

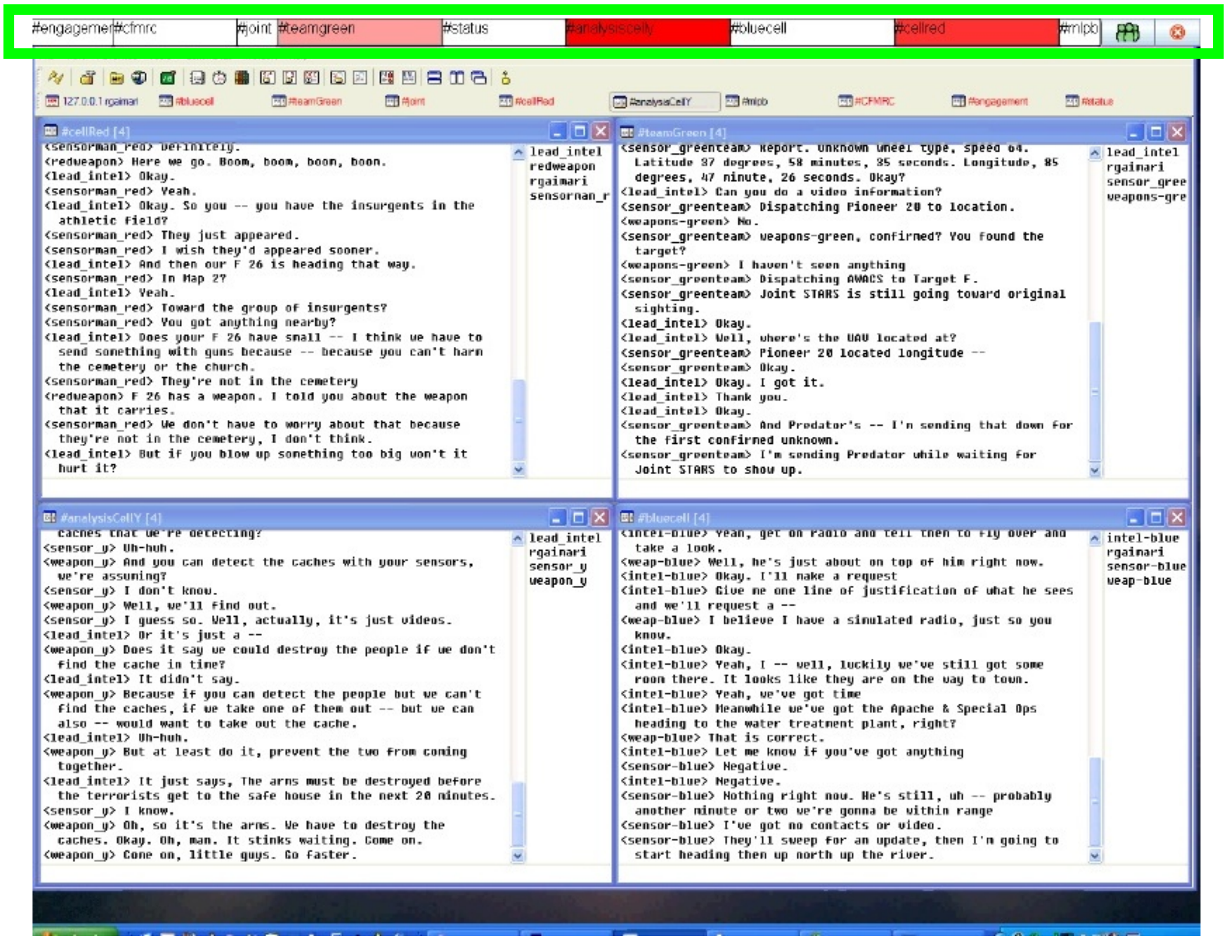

Figure 2. Dynamic Chat Manager Design Concept. (Ex: Monitoring 9 rooms, yet only 4 primary rooms are open.)

Not mIRC Dependent

The Dynamic Chat Manager is not reliant on the use of mIRC as the chat client. We have developed another version of the DCM prototype using a Jabber client, Spark—an open source, cross-platform Jabber/XMPP instant messaging client written in Java[1]. This Jabber client functions somewhat differently, displaying multiple chat rooms via tabbed overlapping windows, such that operators click on a tab to select the desired chat room for viewing rather than viewing many at once. Given this interface, we omitted the DCM display bar and instead modified the built-in tabs to display our cues. Each chat room's ***tab size*** now grows or shrinks to reflect messaging *activity level*, analogously to the mIRC version. Each chat room's ***tab outline color*** reflects the degree of social *network activity*, getting thicker and more intense with more contributions from those in the operator's social network.

TAILORING OF DCM

Users may tailor several features of DCM via a simple interface. An operator can specify unique content alerts that will be integrated into his DCM display. These are specified via the "Alert Keywords" box below (Figure 3) and can be modified at any time. They may select the color used to cue contributions from members of their social network.

---

[1] We utilized the Spark client in conjunction with the Openfire server (previously known as Wildfire), a Jabber/XMPP server written in Java and dual-licensed under both a commercial license and the GNU General Public License.

Users may also tailor their social network defaults, specifying who should initially be included in their network and how strong to make the links. (By default the social networking algorithm will "learn" from monitoring the chat rooms over time and adjust the strengths of an operator's social network links accordingly.) Individuals may wish to prevent a link from decaying over time, as in the case of a user who knows she will not exchange messages often with the Army Liaison yet wants him to remain in her network so she is alerted to his contributions. In such an instance, the operator can specify the strength for the link to Army Liaison and change the setting from ***learn*** (see green "check mark' icon, Figure 3) to ***locked*** (see yellow "padlock" icon, Figure 3). Operators may also delete a connection within their social network, which blocks that link from being "learned" and re-added without operator permission.

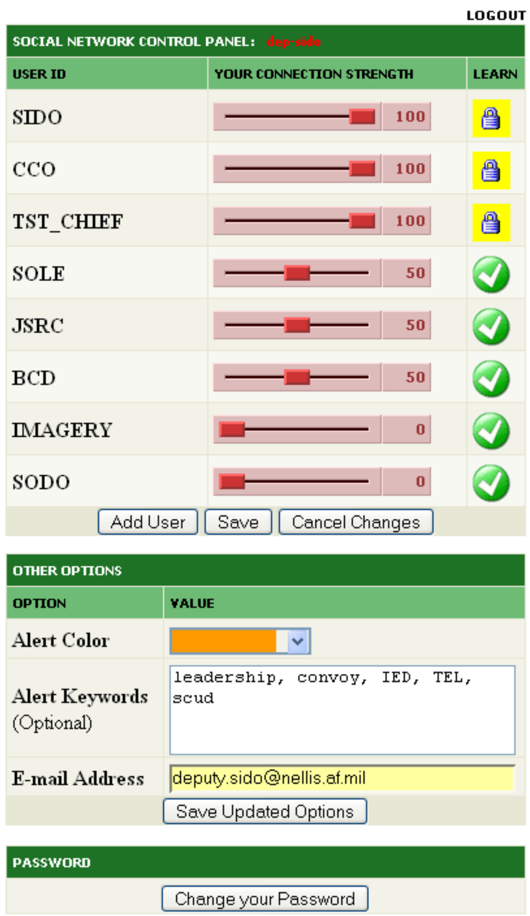

Figure 3. DCM User Interface

Advanced Content Extraction

Berube and Hitzeman (Berube et al, 2007) have developed an information extraction model to determine when specific *content categories* occur within text chat. Unlike simple keyword detection, these models can identify references to more complex classes of content (such as coordinates, targets, or places of interest) within chat and can display very detailed alerts on chosen content categories, color-coding the relevant text within chat rooms.

To explore the potential of this more advanced content extraction, we joined forces to create a "mashup" between their entity extraction capabilities, DCM's chat activity level and social network detection capabilities, and DCM's simple visual interface. The mashup used the Jabber client with its tab-based format. As before, tab sizes change to reflect chat message activity rates, and tab outline color intensity changes to reflect social network activity.

The new feature is a color-coded "content category" icon that appears on the tab of a given chat room if desired content is detected. Within that chat room, the relevant text appears highlighted in the same color for quick location. Operators select the content categories relevant to their role. The integrated system is shown in Figure 4.

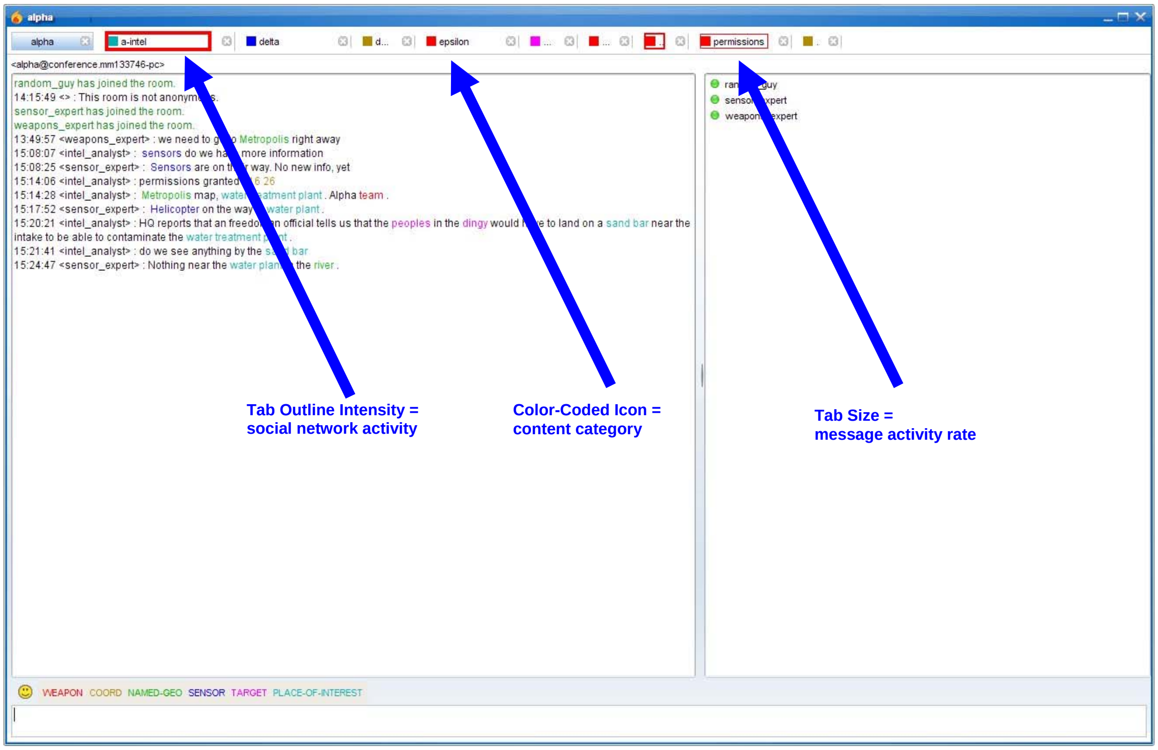

Figure 4. Content Extraction Integrated with Dynamic Chat Manager (Jabber client)

Needless to say, smarter entity extraction (versus basic keyword detection) comes at a price; use of this additional capability requires pre-establishing the content categories for a domain and training the information extraction system on realistic data so that it will recognize appropriate entities (coordinates, targets, places of interest, et cetera). We believe there will be specific sponsor applications in which this additional investment of time and domain knowledge-building will pay significant dividends.

However, we are currently using the basic keyword search within DCM so that users can change their relevant keywords at will, with no system "learning" of a domain required. This simpler approach suits our ongoing evaluations of the technology since users have maximum flexibility and can start using DCM in their mission environment without startup costs or domain-specific constraints. DCM's activity level and social network chat cues are domain independent.

EVALUATION OF DCM

DCM's functionality, capabilities and usability have been evaluated through a limited number of controlled experiments in which teams of three used chat with the Dynamic Chat Manager to respond to a challenging fixed scenario in which they had to collaborate, without any face to face contact, in order to detect and engage six time sensitive targets. While our test subjects did indeed report DCM helpful in directing their attention and reducing the time necessary to detect important information, we do not consider the experimental results to be a formal validation of the technology. The experimentation was extremely helpful as a more qualitative and formative evaluation and as an opportunity for feedback that led to enhancements of the prototype. Several subsequent demonstrations with highly experienced subject matter experts also provided strong validation for the usefulness of the concepts, and feedback on ways to further improve the prototype.

The Dynamic Chat Manager has been selected for operational testing and subject matter expert evaluation during a current operational exercise at an experimental CAOC. We hope to have additional subject matter expert evaluations and feedback from this exercise by the time we discuss this conference paper.

CONCLUSION

Our technology reduces the number of distinct information streams (chat rooms) operators need to fully display by providing summary cues to room activity levels, key participants and chat content. This reduces operator overload and confusion, while enhancing their ability to manage collaboration and rapidly detect important emerging threads. These enhancements to the collaborative environment are directly based on documented difficulties in real-time distributed collaboration environments. As chat becomes increasingly ubiquitous, there is growing need for supporting tools such as these to help reduce team overload and distractions, lessen the risk of missing key information, and facilitate faster and more agile decision-making.

ACKNOWLEDGEMENTS

This work was supported by the Mission Oriented Investigation and Experimentation (MOIE) Program under the Air Force Contract (FA8721-07-C-0001) in MITRE's C3I Federally Funded Research and Development Center (FFRDC). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the sponsor.

REFERENCES

1. Bennett, K. and Flach, J. (1992) Graphical displays: Implications for divided attention, focused attention, and problem solving, *Human Factors*, vol. 34, 513–533.
2. Berube, C., Hitzeman, J., Holland, R., Anapol, R. and Moore, S. (20007) Supporting Chat Exploitation in DoD Enterprises. *12th International Command and Control Research and Technology Symposium (CCRTS) Proceedings,* Newport, RI.
3. Boiney, L. (2007) More than Information Overload: Supporting Human Attention Allocation, *12th International Command and Control Research and Technology Symposium (CCRTS) Proceedings,* Newport, RI.
4. Boiney, L. (2005) Team Decision Making in Time-Sensitive Environments, *10th International Command and Control Research and Technology Symposium (CCRTS) Proceedings,* McLean, VA.
5. Endsley, M. R. (2000) Theoretical underpinnings of situation awareness: A critical review, In Endsley, M. R. and Darland, D. J. (Editors) *Situation Awareness Analysis and Measurement*, Mahwah, NJ.
6. González, Victor and Mark, Gloria. Constant, Constant (2004) Multi-tasking Craziness: Managing Multiple Working Spheres *CHI,* Vienna, Austria.
7. Hudson, James, Christensen, Jim, Kellogg, Wendy A. and Thomas Erickson (2002) I'd Be Overwhelmed, But It's Just One More Thing to Do: Availability and Interruption in Research Management. *CHI,* Minneapolis, Minnesota.
8. McCrickard, D. S. and Chewar, C. M. (2003) Attuning notification design to user goals and attention costs, *Comm. of the ACM* 46, 3, 67-72.
9. Mitchell, P. J., M.L. Cummings, T.B. Sheridan (2004). Human Supervisory Control Issues in Network Centric Warfare. Report HAL2004-1 *Massachusetts Institute of Technology, Humans and Automation Laboratory,* Cambridge, MA.
10. Shneiderman, Ben (1992) Tree visualization with tree-maps; 2-d space-filling approach. *ACM Transactions on Graphics (TOG),* Vol. 11, No. 1.