HUMAN-MACHINE ARTICULATION WORK: FUNCTIONAL DEPENDENCY DIALOGUE FOR HUMAN-MACHINE TEAMING

Clayton D. Rothwell
Infoscitex Corporation and Wright State University
Dayton, Ohio
Valerie L. Shalin
Wright State University
Dayton, Ohio

Articulation work is an overlooked requirement for successful human-machine teams. Articulation work captures the often hidden task management activities human-human teams regularly perform in response to functional dependencies amongst team members. While human-human teams demonstrate articulation work through language, human-machine teams currently do not. Aviation is replete with examples, from the superficially mundane adaptation inherent in the turnaround of commercial aircraft to the life-threatening misunderstanding in Turkish Airlines Flight 1951 and Asiana Flight 214. Current research in human-machine language-mediated interaction has failed to study tasks that are sufficiently complicated to require articulation work, resulting in a misleading optimism about the state of the art. More realistic scenarios in human machine teaming will promote attention to this fundamental limitation and motivate the development of analogous capability.

For several decades, initiatives such as crew resource management acknowledge team processes as critical to performance and safety. As both military and commercial aviation evolve in a stream of technological advances, human-machine teams are a new possibility and goal. Legacy frameworks for designing automation-based technologies are a natural starting point for human-machine teams. However assumptions of a static environment make these frameworks inherently rigid and brittle. In contrast, human-human teams are particularly fluid in the dynamic task management activities known as *articulation work*, typically accomplished through natural language dialogue. While natural language technology could enable human-machine articulation work, current technology assumes overly simplistic tasks and notions of cooperation, omitting articulation work as a requirement. This position paper argues that 1) articulation work is critical to the success of human-machine teams just as it is in human-human teams, 2) frameworks for designing human-machine teams do not allow for sufficient articulation work, 3) current language technology does not support articulation work, 4) requirements for articulation work using dialogue must drive new research and development.

The Importance of Articulation Work

Cooperative work in a dynamic environment necessitates articulation work, that is, task management activities aimed at functionally decomposing a task, negotiating goals, identifying dependencies, and divvying up who will do what and when. Articulation work establishes the team's varying *functional dependencies*, that is what team members are committed to doing and what other teammates are depending on them to do. Schmidt and Simone (1995) note that articulation work also serves to improve collaborations as team members:

> tacitly monitor each other; they perform their activities in ways that support coworkers' awareness and understanding of their work; they take each others' past, present and prospective activities into account in planning and conducting their own work (p. 17)

Articulation work obtained prominence in research areas focusing on teamwork and team performance, such as computer supported cooperative work (CSCW; Schmidt and Bannon, 1992; Malone and Crowston, 1990). Importantly, the articulation for a particular situation is not rigid, it's adapted as members leave and join the group, become fatigued over time, demonstrate competence or incompetence, learn new capabilities, or deplete certain resources. The hallmark of articulation is how it enables robust teamwork in the face of the moment-by-moment unexpected—uncertainty and environmental perturbations that cause a collision between the team's plans and execution. Teams discuss the nature of perturbations, their existing plans and commitments, and discuss whether or not the perturbations merit a change in approach from one or more members.

Consider what a team of airport and airline employees must face while performing turnarounds, the process of taking a plane that's just arrived, unloading and servicing it, and loading it again so it can take off again (Wales, O'Neill, Mirmalek, 2002). Articulation work allows a team to discuss perturbations as well as existing constraints and determine how to respond in concert, continuing the interdependent activities required to meet the goals. While the flight schedule determines resourcing staff and equipment, perturbations emerge in the form of flight delays, changes and cancellations due to weather and mechanical problems. When a flight is canceled and passengers are moved to another plane by operations personnel, bag handling personnel have to accommodate these changes. Turnarounds take longer or shorter than expected, spawning gate changes. Staff members report in ill or equipment breaks, such as fuel or catering trucks. On close inspection the plan *never* unfolds exactly as anticipated, but (hidden) articulation work fills in the gaps, including gaps in the technology, to create the misleading impression of (mostly) seamless integration.

Historical Human-Machine Frameworks

Legacy frameworks for designing automation-based technologies are a natural starting point for human-machine teams. Fitt's (1951) list men-are-better-at, machines-are-better-at (MABA-MABA) approach seeks to divide responsibility between humans and machines. It accomplishes this by identifying the (relatively) superior capabilities of humans and machines and then allocating tasks to whomever or whichever is the most proficient. Allocations suggested in the Fitt's list have been commonly understood as static assignments (see de Winter & Dodou 2011 for discussion). Later researchers identified the MABA-MABA approach as overly simplified and sought to have multiple levels of automation that can be isolated for different stages (e.g., Parasuraman, Sheridan, & Wickens, 2000). However, the levels of automation framework remains rigid. These use static task decompositions rather than allowing for articulation work. Dynamic function allocation, adaptive automation with machine-initiated changes, and adaptable automation with human-initiated changes (for a review see Scerbo, 1996), though they provide for change, are insufficient. These approaches have predefined the possible changes, the notifications of change, and the triggers for changes. Consistent with

Norman (1996), fine-grained articulation work is always necessary to generate novel team structures or distribution of responsibilities, provide flexibility in how to notify team members of change, and use triggers that can't be predicted or exhaustively programmed. Moreover, changes in automation status are poorly communicated to human teammates and therefore often missed, leading to mode confusion and automation surprise (Sarter, Woods & Billings, 1997). While Woods (1996) recognizes the expansive consequences of new automation on the distribution of responsibility, the molar time-scale of work practice adaptation neglects the moment-by-moment adaptation that teams require. Coactive design is a relatively new approach that seeks to detail human-machine interdependencies (Johnson, Bradshaw, Feltovich, Jonker, van Riemsdijk, & Sierhuis, 2014), however its breadth lacks specific guidance for dialogue system development.

Functional Dependencies in Aviation Accidents

Machines in use are currently inept at articulation work. Typically, expert humans resort to workarounds to distribute the functions amongst system members while managing the tasks supposedly distributed to the machine. Two similar commercial aviation accidents support this assertion: Turkish Airlines Flight 1951 and Asiana Flight 214. In each of these accidents, the humans were depending on the machine to perform a function, the machine was not aware the human was depending on it and the humans were not aware of the machine's tacit decline of responsibility. In both accidents, the critical function was to maintain thrust on approach through the autothrottle.

Turkish Airlines Flight 1951. On 25 February, 2009, Turkish Airlines Flight 1951 crashed during its approach to the Amsterdam Schiphol airport. The first officer as pilot flying was using Line Flying Under Supervision, which utilizes the autothrottle for airspeed control. Though the pilot flying and the crew were relying on the autothrottle to maintain the airspeed of the aircraft on the approach, the aircraft could not be informed of this functional dependency. Rather, the flight crew attempted to create the airspeed function through management of the autothrottle mode selections. The approach was higher than the glidepath, so a member of the flight crew selected the 'vertical speed' mode to increase the descent. After this change, the autothrottle entered RETARD mode, which was displayed on the left and right primary flight displays, and the autothrottle moved the thrust levers into the idle position. In contrast to crew assumptions, the autopilot would not maintain airspeed in this configuration. However, the machine's exclamation of RETARD to the flight crew was unspecific and did not communicate the breakdown of the expected function. There are two types of RETARD, one for flight level changes and one for flaring to land, and the primary flight display annunciation panel does not distinguish between the two (Silva and Hansman, 2015). The type of RETARD depended on the altitude information reaching the autopilot, with a threshold altitude of 27 feet. The autopilot believed the aircraft to be below 27 feet in altitude because the autopilot was receiving and using erroneous altitude data indicating the aircraft height at -8 feet, which disagreed with the altitude data presented to the pilot flying. The pilot's primary flight display showed a conflicting but correct altitude status, leading to confusion over the situation. Ultimately, the lack of clear functional dependency and the misunderstanding about the meaning of RETARD led to an unrecoverable stall and the aircraft crashed killing 9 and injuring 117 (Dutch Safety Board, 2010).

Asiana Flight 214. An accident of similar origin occurred when Asiana Flight 214 crashed on July 3rd, 2013 during approach to San Francisco International Airport. Asiana has an informal practice for visual approach of turning off both flight directors and then turning back on the pilot monitoring's flight director during the approach (National Transportation Safety Board, 2014). This practice results in the autothrottle entering speed mode and determines a distribution of functions: the autothrottle maintains airspeed and the pilot flying can focus on pitch and roll. In Flight 214, the aircraft was above the glidepath and needed to descend. The informal practice of toggling the flight director was followed 'loosely,' both flight directors were not off at the same time and therefore speed mode was not entered. The pilot flying moved the thrust levers and inadvertently caused the autothrottle to change to HOLD mode. The HOLD mode created a breakdown in the function of maintaining airspeed—HOLD deactivated automatic airspeed control. The burden of recognizing the mode change and the implications for the function being provided in the approach falls on the flight crew, but they did not note the change to HOLD mode. These events led the aircraft to descend below the glidepath at a high rate and collide with a sea wall, killing 3 and injuring 187 (National Transportation Safety Board, 2014).

Both cases hinge on the absence of human-machine articulation dialogue concerning the retention or abandonment of otherwise tacit commitments to act. These problems with managing functional dependencies are relatively simple when compared to the envisioned applications for human-machine teams.

Implications for Human-Machine Communication

Human teams routinely perform articulation work through dialogue. Research into human-machine communication and natural language dialogue has exerted a great deal of effort studying and improving clarification of utterances or lexical ambiguities, but not communication, clarification or negotiation of functional dependencies, which require a more complicated ontology including agent beliefs (e.g., Clancey, Sierhuis, Damer, & Brodsky, 2005). Many of the classic human-machine communication tasks do not provide opportunity for articulation work and therefore do not reveal these deficiencies.

Common application domains for natural language processing, such as shopping and navigation, are restricted and fail to address the ways in which these tasks vary. As the scope of artificial intelligence grows and natural language processing technologies become more integrated into work practices, their applications will not be limited to the subset of activities with overly simplified team processes. When a richer task is used, such as a collaborative problem-solving task similar to the board game Clue (Traum & Dillenbourg, 1996), the proportion of communication spent on articulation work is apparent. In particular, the authors noted a frequent topic was decomposition of who does what and when. Simple tasks lead to the impression that a simple ontology can work, whereas a moderately complex task (still quite simple in comparison to the wild) easily sets a high bar for a rich ontology comprising not just the task content but the possible conceptualizations and organizations of the cooperation. Research is needed to push dialogue-mediated tasks into more realistic scenarios that will require articulation work.

Future Research

The persuasive macro-level case for articulation capability does not provide concrete requirements for designers and software developers for building natural language interfaces. We specify three topics that should shape the research agenda.

Initiate team research utilizing task settings that require articulation. A primary goal of future research should be to have some aspect of the task that prompts articulation work. We envision articulation work to be prompted by an element of dynamic disparity in the task context that differs between partners and thus affects the rate of progress for one of the partners, which merits announcing to or discussing with the other partner. This could be due to a change in a sub-goal of the task, which requires discussion of a change in strategy, approach, sequence, or the like. This could also be due to introducing a problem that perturbs the existing strategy.

Formalizing a taxonomy of articulation work. A taxonomy is needed to enable diagnosis of the disconnects between human and machine teammates. Research programs from J. Allen, H. Clark, and J. Searle provide theoretical inspiration. A critical requirement is a conceptual distinction between real-time execution and planning activities (Shalin, 2005). It is the interaction between execution and planning, frequently initiated by a perturbation, that spawns articulation dialogues. While resources such as Aviation Safety Reporting System and specific accidents provide data, what is required is the conceptual framework to generalize the limitations, across instances in aviation and ideally, extending to other domains including laboratory tasks. Being accountable for providing a specific function is one facet of this taxonomy. Terminology grounding (e.g., which meaning of RETARD) and explicit task completion acknowledgement (as in toggling the flight director) is another.

Translate articulation requirements into functional machine analogues. Attempting to replicate human team members with machines is fraught with philosophical problems and an impractical near term goal at best. Nevertheless specific functions are well within technical reach without imbuing technology with human processes, e.g., for clarifying the grounding of terminology, confirming mode change communications are received, identifying that a functional dependency will not be upheld. These functions will drive the requirements for natural language interfaces.

Conclusion

In this position paper, we have argued that 1) articulation work is critical to the success of human-machine teams just as it is in human-human teams, 2) frameworks for designing human-machine teams do not allow for sufficient articulation work, 3) current language technology does not support articulation work, 4) requirements for articulation work using dialogue must drive new research and development. Human-machine teaming research to date has largely ignored the challenge of articulation work and cannot ignore it any longer.

Acknowledgements

This research was partially supported by the Air Force Research Laboratory's Human Interface Research Technologies Contract #: FA8650-14-D-6501. We thank Asaf Degani and Immanuel Barshi for helpful discussions early on in the development of this work.

References

Clancey, W.J, Sierhuis, M., Damer, B., Brodsky, B. (2005) Cognitive modeling of social behaviors. In R. Sun (Ed.), *Cognition and Multi-Agent Interaction: From Cognitive Modeling to Social Simulation*, (pp. 151-184). New York: Cambridge University Press.

Dutch Safety Board (2010). Crashed during approach, Boeing 737-800, near Amsterdam Schiphol Airport, 25 February 2009. *Dutch Safety Board Final Report*.

Fitts, P. M. (ed) (1951) Human engineering for an effective air navigation and traffic control system. National Research Council, Washington, DC

Johnson, M., Bradshaw, J. M., Feltovich, P. J., Jonker, C. M., Van Riemsdijk, M. B., & Sierhuis, M. (2014). Coactive design: Designing support for interdependence in joint activity. *Journal of Human-Robot Interaction*, *3*(1), 43-69.

Malone, T. W., & Crowston, K. (1990, September). What is coordination theory and how can it help design cooperative work systems?. In *Proceedings of the 1990 ACM conference on Computer-supported cooperative work* (pp. 357-370). ACM.

National Transportation Safety Board (2014). Crash of Asiana Flight 214 Accident Report Summary.

Norman, D. A. (1992). Coffee cups in the cockpit. In *Turn Signals are the Facial Expressions of Automobiles*, (pp. 154-174). Boulder, CO: Westview Press.

Parasuraman, R., Sheridan, T. B., & Wickens, C. D. (2000). A model for types and levels of human interaction with automation. *IEEE Transactions on systems, man, and cybernetics-Part A: Systems and Humans*, *30*(3), 286-297.

Sarter, N. Woods, D. & Billings, C. (1997). Automation surprises. In G. Salvendy (Ed.) Handbook of Human Factors & Ergonomics *2*, (pp. 1926-1943). Hoboken, NJ: Wiley.

Scerbo, M. (1996). Theoretical perspectives on adaptive automation. In R. Parasuraman & M. Mouloua (Eds). *Automation and human performance: Theory and applications*., (pp. 37-63). Hillsdale, NJ: Lawrence Erlbaum.

Schmidt, K. and Bannon, L. (1992). Taking CSCW seriously: Supporting articulation work. *Computer Supported Cooperative Work*, 1(1), 7–40.

Schmidt, K. and Simone, C. (1995). Coordination mechanisms: An approach to CSCW systems design. In Navaro, L. (Ed). Demonstrator prototypes of computational mechanisms of interaction, (pp. 15-72). Lancaster, UK: Lancaster University.

Shalin, V. L. (2005). The roles of humans and computers in distributed planning for dynamic domains. *Cognition, Technology & Work*, *7*(3), 198-211.

Silva, S. S. and Hansman, R. J. (2015). Divergence Between Flight Crew Mental Model and Aircraft System State in Auto-Throttle Mode Confusion Accident and Incident Cases. *Journal of Cognitive Engineering and Decision Making*, 9(4), 312–328.

Traum, D., & Dillenbourg, P. (1996, August). Miscommunication in multi-modal collaboration. In *AAAI Workshop on Detecting, Repairing, And Preventing Human-Machine Miscommunication* (pp. 37-46).

Wales, R., O'Neill, J., & Mirmalek, Z. (2002). Ethnography, customers, and negotiated interactions at the airport. *IEEE Intelligent Systems*, *17*(5), 15-23.

de Winter, J. & Dodou, D. (2011). Why the Fitts list has persisted throughout the history of function allocation. *Cognition Technology & Work*, *16(1), 1-11.*

Woods, D. D. (1996). Decomposing automation: Apparent simplicity, real complexity. In R. Parasuraman & M. Mouloua (Eds). *Automation and human performance: Theory and applications*, (pp. 3-17). Hillsdale, NJ: Lawrence Erlbaum.