Strategic Conformance: An Important Concept In Future Automation Design?

Brian Hilburn
Carl Westin
Center for Human Performance Research
Spinozalaan 34
2273XC Voorburg, the Netherlands
+31 6 1227 6979
brian@chpr.nl
carl@chpr.nl

Clark Borst
Delft University of Technology
Control and Simulation Department
Delft, the Netherlands
c.borst@tudelft.nl

Abstract-- The MUFASA project simulated sophisticated ATM conflict resolution automation using unrecognizable replays of controllers' own performance. Using a prototype air traffic control interface, the project explored with operational air traffic controllers the interactive effects of traffic complexity, level of automation and "strategic conformance" (defined as *the match between human and machine solution strategy*) on a number of dependent measures. Conformance was found to impact acceptance, agreement, and response time. Conformal advisories were accepted more often, rated higher, and responded to faster than were non-conformal advisories. In the end, one result stood out in particular: roughly 24% of conformal advisories were rejected by controllers. How could it be that controllers, in effect, disagreed with their very own solutions roughly one quarter of the time? The project is currently exploring this and other related issues through extended human-in-the-loop simulations.

Keywords- Automation, ATM, Acceptance, Strategic conformance

I. INTRODUCTION

Roughly 60 years ago, English mathematician Alan Turing famously posed the ultimate test for artificial intelligence: that its performance be indistinguishable from that of a human. If one could converse with an unseen agent, and mistake computer for human responses, then that computer could truly be said to "think." This notion has driven research into artificial intelligence for over half a century.

We are at a point in the evolution of automation that we routinely turn over to computers many of the "thinking" tasks previously performed only by humans. Our planes, trains and even automobiles rely on more, and more capable, automation than ever before. Despite various achievements, however, there remains some gap between the theory and practice of automation design. For instance, as of this date not a single computer has passed the Turing test. Nor have we realized in any meaningful way the highest levels of autonomous systems. As Sheridan noted, we still have no idea how to program computers to "take care of children, write symphonies or manage corporations…." [1]

The ongoing MUFASA (Multi-dimensional Framework for Advanced SESAR Automation) project started, in a sense, from the opposite view: What if we could build perfect automation, which behaved and solved complicated problems exactly like a human? Would the human accept its advice? Or might humans reject solutions simply because these were offered by automation? The question is whether there is evidence of a fixed bias against automation, irrespective of its performance.

These questions are not just academic. The MUFASA project started from the assumption that future ATM will increasingly rely on automation that can assume control of the cognitive and strategic aspects of ATM. The SESAR target concept for future Air Traffic Management (ATM) is built on an evolutionary path of five "Service Levels" that correspond to progressively more sophisticated automation. Successfully introducing such advanced new forms of automation might rely heavily on initial air traffic controller acceptance, at least for some transitional period. It was therefore reasonable to hypothesize that controller acceptance, at least initially, would rely on the machine working in a way that was familiar to the human. We captured this notion in the concept of *strategic conformance*, which we defined as

> *…the degree to which automation's behavior and apparent underlying operations match those of the human.*

MUFASA laid out an initial predictive framework for automation usage and acceptance, and set out to explore how these would be impacted by the possibly interactive effects of three factors:

- Traffic complexity;
- Level of automation; and
- Strategic conformance.

A. *Research questions*

The main aim of the MUFASA project was to investigate the possibility that controllers would show a systematic bias against automation, which could jeopardize the introduction of advanced forms of ATM automation. Specifically, would controllers be accepting of automation that is designed to replace aspects of their strategic decision-making in the areas of conflict detection and resolution, even if the automation solves problems in the same way as the human? This question draws inspiration in part from EUROCONTROL's CORA project [2], which recognized the potential benefit of *heuristic* automation, which aims to solve problems in the same way as humans.

Specific research questions addressed in the MUFASA project included the following:

- Are controllers more likely to accept automated advisories when these mimic the controller's own solution?
- Everything else equal, does acceptance vary by air traffic complexity?
- Do other measures (such as workload ratings, or response time) show an impact of strategic conformance?

Previous attempts to simulate such automation have been hindered by one important limitation: they generally could not guarantee that simulated automation performed a high level task in exactly the same way, at least outwardly, as does the human. MUFASA started with a fairly novel experimental design and simulation protocol, which allowed us to capture and replay (in an unrecognizable way) specific ATM scenarios, including controller's specific actions. This allowed us to ask the following intriguing question:

> *Do controllers reject their own previous solutions, when they (mistakenly) believe that these come from automation?*

II. METHODS

We conducted a series of two human-in-the-loop simulations: The first captured air traffic controllers' manual performance in maintaining safe separations between aircraft; The second replayed for them "automated solutions" which were in fact either unrecognizable replays of their own previous performance (the "conformal" condition) or replays of a colleague's different but acceptable solution (the "non-conformal" condition). Controllers were free to either accept a given advisory, or to reject it and implement an alternative solution.

The simulation was based on a modified prototype of the Solution Space Diagram (SSD, Figure 1), which is currently under development at the Delft University of Technology. The SSD is a tactical decision support tool that displays color coded GO and NO-GO vectors to facilitate controllers' use of speed and heading resolutions [3]. As shown in figure 1 (converted from color to gray scale here) the aircraft (on a heading of roughly 290) is heading to a medium term conflict (yellow in the SSD). A turn of either say 20 or 80 degrees to the right would put it on a conflict-free trajectory (dashed green), whereas a 180 turn would result in a short term conflict (red).

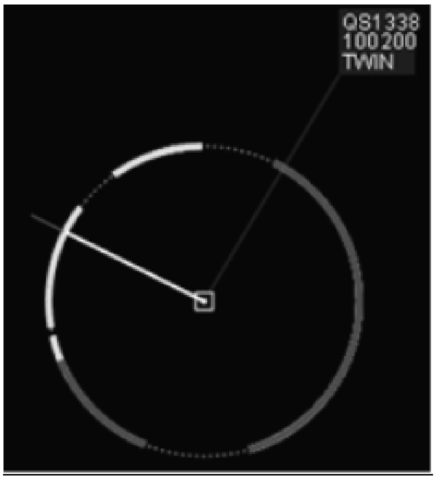

Figure 1. Close-up of the SSD screen showing GO, NO-GO short term conflict and NO-GO medium term conflict headings.

A. Participants

Sixteen professional air traffic controllers voluntarily participated at the Shannon Area Control Centre, Ireland. Experience ranged from zero to ten years (mean = 2.5). Twelve controllers currently worked en-route and one the tower position. Three were en-route students.

B. Equipment

The Java-based simulation ran on a portable computer connected to an external 21" monitor. Participants interacted with the simulator through an externally connected computer mouse and keyboard. The ATC simulator was a Java-based application that allowed air traffic controllers to control short traffic scenarios. To vector an aircraft, a controller used a computer mouse to click on an aircraft of interest, drag the velocity trend vector to a new conflict-free area on the heading ring (a "clear" area outside the red/yellow areas), and press the ENTER key on a keyboard to implement the vector. Speed clearances (and combined speed and heading clearances) could also be given by using the mouse scroll wheel to either increase or decrease speed.

C. Traffic scenarios

Each simulation session consisted of 16 traffic scenarios (each a two-minute vignette of level en-route traffic). All used a squared airspace equal in size. Four baseline scenarios were each rotated in different angles to create three variants, resulting in four scenario groups with four scenario variants in each group. This reduced potential confounding factors, and ensured that initial complexity was the same across scenarios, facilitating comparison between low and high complexity conditions. We maintained sector geometries through scenario rotations in which the relative trajectories and closure angles of aircraft were kept constant, but the entire sector was rotated, and sector entry/exit points renamed.

Each scenario featured only one designed conflict (and this was always between two aircraft). Geometry of the designed conflict was only varied between baseline scenarios. The conflict pair was initially aligned to the exit points and thus required no initial controller interaction. The other aircraft in the sector were considered “noise” aircraft to distract the controller from the conflict pair. Some noise aircraft were misaligned with their exit point and displayed in grey, whereas aligned noise aircraft were displayed in green such that the controller could immediately see which aircraft had not yet been cleared to their respective exit point. Designing the conflict scenarios took a great deal of effort to maintain experimental control (we did not want, for instance, controllers to solve conflicts earlier than the advisory, nor for noise aircraft to disrupt the designed conflict).

D. *Experimental design*

We used a 2x2x2 design, with Conformance, Complexity and Level of Automation (LOA) all varied within subject (Conformance was not fully crossed, as it is only relevant under automated conditions). We discovered during early developmental testing that “conformance” is not a binary measure, and in the end we defined it in terms of aircraft choice, clearance type (e.g., heading change only), and clearance direction (e.g., heading change to the left). A non-conformal solution therefore always featured a different aircraft choice and/or clearance type and direction. Again, non-conformal solutions were always derived from solutions provided by other controllers. Complexity was varied through means of aircraft count, and calibrated in a series of test trials. Finally, presentation order of traffic complexity and solution conformance was balanced between participants and traffic scenarios using a Latin Square design.

Dependent measures included

- Acceptance of an advisory (binary yes or no);
- Agreement with an advisory (on a 1-100 scale);
- Response time (from advisory onset to accept / reject button press), and
- Subjective workload (on a similar 1-100 scale).

E. *Procedures*

The entire simulation lasted four weeks. In the first week, the initial experiment was conducted to capture controller resolutions to the designed conflicts. Following briefing and consent procedures, we conducted 16 training runs and 16 measurement runs. Participants were given two main tasks: resolving conflicts and clearing aircraft to their intended exit point. A continuously updated performance score reflecting these two task parameters was included to keep participants focused and motivated, and more importantly to prevent scenario recognition and early detection of the designed conflict. To warn the controllers for short-term conflicts, an auditory alert was triggered and the aircraft involved in the conflict were displayed in red.

During the middle two weeks, this initial dataset was processed and two sets of automated scenarios (one set of eight conformal advisories, and one set of eight non-conformal advisories) were created for each participant. In the final week, the same 16 controllers again took part, this time by interacting with 16 “automation” scenarios, which had been tailored to them to show eight conformal (again, unrecognizable replay) and eight non-conformal (again, a colleague’s different solution to the same scenario) resolutions. Participants performed the same task as in the earlier experiment, but now were assisted by a higher level of automation that would occasionally provide resolution advisories by proactively auto-selecting a conflict aircraft.

The resolution advisory consisted of a heading vector, a speed vector, or a combination thereof. The resolution advisory was accompanied by a beeping sound and a dialog window that the controller used to either ‘accept’ or ‘reject’ the advisory. Accepting the advisory would immediately implement it, whereas rejecting it would extinguish the displayed solution and leave the controller free to implement an alternative solution. There was a 15 second timeout on the solution. Participants were told that an advisory would always solve the conflict, but not necessarily in the most optimal way, and were therefor encouraged to find their own preferred solutions.

After each scenario, participants were given performance feedback in terms of an average performance score. Second, controllers were asked to rate subjective workload and their agreement with the automation advisory.

III. RESULTS

A. *Acceptance*

Overall, controllers accepted 340 of 512, or 66%, of all advisories. Both complexity and conformance showed a significant main effect on acceptance rate. Controllers accepted roughly 75% and 56% of advisories under high and low complexity conditions, respectively. Conformance showed a nearly identical effect: with controllers accepting 76% and 57% of advisories under conformal and non-conformal conditions, respectively. There was also a very weak interaction trend (p=.37) on acceptance: controllers tended to show higher acceptance for conformal solutions, and this effect was more pronounced under high complexity conditions (figure 2).

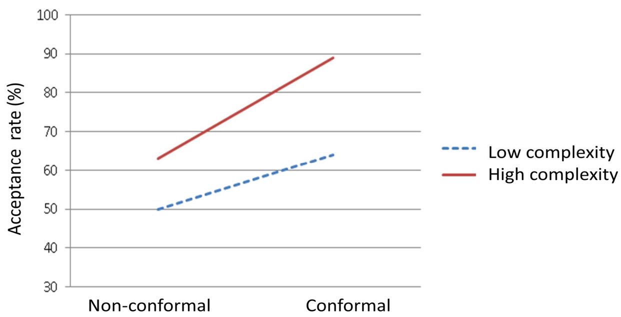

Figure 2. Acceptance rate (percent), by Complexity and Conformance.

B. *Agreement with automated advisories*

Regardless of whether a given advisory was accepted or rejected, controllers were instructed to indicate (on a scale of 1-100) their agreement with the advisory immediately after each scenario. Standardized agreement ratings showed a significant main effect of both complexity and conformance. Acceptance ratings were significantly higher for conformal than non-conformal solutions (average *z* score of +.11 and -.11, respectively). Agreement was higher under complex conditions (average *z* score of +.05 and -.05 under high and low complexity). Agreement rating showed a borderline-significant interaction trend (p<.1) between complexity and conformance. Controllers tended to show higher agreement with highly conformal solutions, but this effect was less pronounced under high complexity conditions (figure 3).

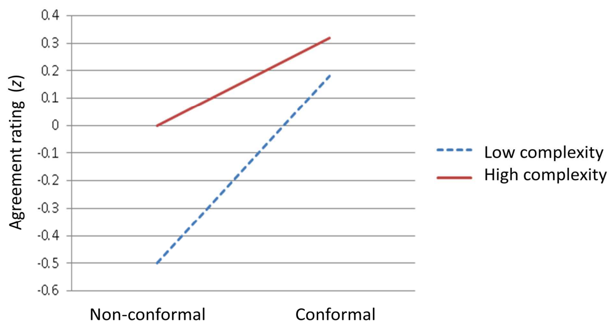

Figure 3. Standardized agreement rating (mean), by Complexity and Conformance.

C. *Response time to automated advisories*

Response time (again, from advisory onset to acceptance / rejection) ranged from approximately 0.6 sec to 14.6 sec (again, there was a 15 sec maximum). Response time showed a significant main effect for conformance, but not for complexity. Overall, response time was significantly higher for non-conformal advisories (5.9 sec vs 4.9 sec): controllers responded faster to conformal advisories. Response time was also higher for low vs high complexity conditions (5.7 sec vs 5.1 sec), though this difference was not significant.

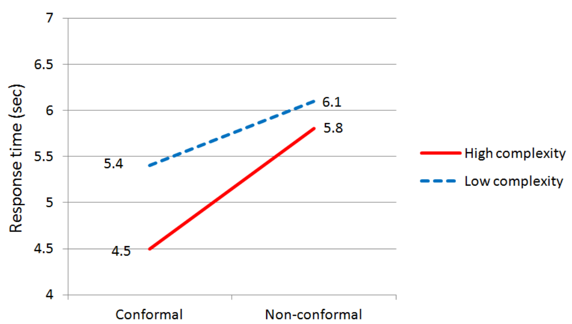

Figure 4. Response time (sec), by Complexity and Conformance.

Overall, response time was higher (i.e. controllers were slower) to reject than to accept advisories, 6.9 sec vs 4.7 sec. Conformance had a significant main effect on response time, with conformal solutions showing associated with higher response times. This effect appeared about the same for both accepted and rejected advisories. Response time decreased with complexity. That is, controllers responded faster to complex conditions.

D. *Workload ratings*

Difficulty ratings were obtained after each session, on a scale of 0-100. Notice that these ratings referred to the entire scenario, not just the advisory. Workload ratings increased with complexity ($F[1,15]=179.95$, $p<.01$). Neither the main effect of conformance, not the interaction between the two, was significant. When we broke out these conformance results by acceptance status (whether an advisory was accepted or rejected), however, a different picture emerged. As shown in figure 5, the influence of conformance on workload rating looked very different for accepted and rejected advisories. It was rated easier when a conformal solution was accepted, and easier when a non-conformal solution was rejected.

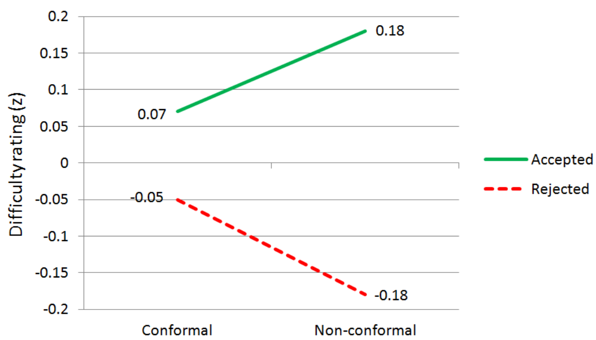

Figure 5. Average workload rating (z), by Conformance and Accept / Reject status.

E. *Debrief interview feedback*

Two main themes emerged from post-session debriefs, one encouraging and one cautionary. First, several controllers noted that the prototype automation made possible a new way of working (a sometimes-desirable result of new automation), by facilitating the use of speed adjustments (which they do not tend to currently use in en route airspace). However, several participants also noted the tendency to feel “driven” by the automation, become reactive, and to curtail their conflict assessment under high complexity situations.

In terms of experimental design, one bit of post-session feedback was gratifying: controllers reported that they had not recognized replays as having been their own previous performance.

IV. DISCUSSION

A main effect of conformance was observed on acceptance, agreement, and response time. Conformal advisories were accepted more often, rated higher, and responded to faster than were non-conformal advisories. Complexity, on the other hand, showed a main effect on acceptance, agreement and workload: all three increased with complexity. Essentially, under complex conditions controllers were more likely to accept their own (versus non-conformal) solutions, yet were also less (but still) prone to dislike non-conformal advisories.

One interpretation of the agreement effect is that controllers, under the time pressure of complex conditions, did not fully evaluate a resolution advisory. Qualitative analysis of controllers' conflict resolution performance indicated that controllers were inconsistent both internally and in comparison to their colleagues. If this is true, it challenges the majority of automation design that follows a "one-size-fits-all" approach. This finding, combined with the observed effects of conformance on acceptance, agreement, and response time have the potential to overcome the acceptance issues and disuse of more advanced automation already observed in current ATM and other domains.

V. CONCLUSIONS

MUFASA has assumed that strategic conformance is an important construct underlying controller acceptance and usage of new automation. Indeed, we found a main effect of conformance on acceptance, agreement ratings, and response time. Conformal advisories were accepted more often, rated higher, and responded to faster than were non-conformal advisories. It has been argued, though, that this might be a short-lived effect and that, in the long run, strategies will shift and "dispositional" trust will hopefully grow.

This might well be the case. In this project we have addressed only the transitional era, during which controllers have to adapt, accept, and transition to new and highly advanced (decision aiding) automation, not the mature future state of automation. However, we question whether there will ever be a truly mature state, as technological advances will probably always drive what we aim to do with automation, and the future will likely remain a moving target. We also feel that the ATM community is on the cusp of a fundamentally new era of human-machine interaction (though perhaps this has always felt so): automation is now capable of giving strategic advice, and becoming a partner / agent in the control room. This represents a qualitative shift and increases the threshold for controller trust. It also means that automation might be more "ignorable" by the controller if its benefits are not recognized.

Which leads to a potential paradox: a controller might have to use a system before he comes to trust it, yet he has to trust it before he will use it. This is why the initial acceptance of automation might be so critical. If we do not get controller buy-in, rollout of automation is much more difficult. .How do we get to a mature state, of dispositional trust and well-behaved automation, without initial acceptance and usage? How do we strike a balance between complacency and skepticism? If, after all, we do not expect our controllers in the future to calibrate trust in a system, and to (at least show the potential for) distrust, we might even ask whether we should keep a human in the loop, or instead completely automate that part of the process.

VI. ONGOING RESEARCH

The MUFASA project developed a simulation platform and unique experimental protocol that allowed us to capture controller performance and replay it in such a way that we could simulate "automation performance" using unrecognizable replays of a given controller's own previous performance. In essence, "automation" was now, for the first time, able to perform exactly like the controller. The project refined this simulation capability over a series of developmental simulations, culminating in a pair of real-time simulations with sixteen air traffic controllers. Simulations explored the interactive effects of strategic conformance, traffic complexity, and level of automation, and their impact on controller acceptance, agreement, response time and workload.

In the end, one result stood out in particular to us: of 256 conformal solutions (i.e. replays of controllers' very own previous performance), 61 (or 23.8%) were rejected by controllers. How is it that controllers would disagree with themselves nearly one quarter of the time? One speculation is that controllers are simply inconsistent over time in the solution and strategy they might choose to employ. Alternatively, it could be that controllers are not necessarily opposed to automation per se, but to advisories from any source (even, say, from a colleague). These and other remaining questions have led us to extend the results presented here, to explore over the coming months the following research topics in additional human-in-the-loop simulations:

A. Controller consistency

We demonstrated that controllers can sometimes disagree with their very own solutions, if they are later presented as those of automation. Could it be, however, that this is merely intra-individual noise? That is, are controllers simply inconsistent over time in the strategies they bring to the conflict resolution task? Previous research into ATC consistency has tended to focus on <u>inter</u>-controller consistency. In terms of <u>intra</u>-controller consistency, data are both sparse and unclear. To address the topic, participants will perform multiple manual (i.e. unaided) runs of the same (unrecognizable) traffic scenario.

B. Source bias

Were participants, when rejecting advisories, actually rejecting advisories, or were they rejecting automation per se? Would controllers be as likely to reject advice, if they believed that such advice had come from a trusted colleague? Addressing this topic involves only a very simple extension of our current experimental protocol. Specifically, we will revise our participant instructions to indicate (within subject) either colleague or automation as the source of the advisories. Experimental design remains otherwise unchanged.

C. Automation transparency

For reasons of experimental control, our simulations thus far have maintained a fixed level of automation transparency. However, one could plausibly speculate that the "opacity" of the SSD tool itself might have driven controllers to sometimes reject conformal advisories. Were controllers rejecting some of the advisories because the SSD allowed them to see alternative solutions to a pending conflict? In general, higher levels of automated support are associated with decreasing human involvement in the decision-making process. In most studies that involve machines with higher levels of autonomy and authority, less attention is paid to the human-machine interface in terms of providing information. It has been argued, however, that interacting with smart technologies actually requires more information (i.e., richer interfaces), not less [4]. Thus far, there does not seem sufficient evidence to draw a conclusion, and we look forward to exploring this, as well as the previously mentioned topics, in the months ahead.

ACKNOWLEDGEMENTS

This work was supported under the SESAR Innovative Research Programme (*Workpackage E*), and by EUROCONTROL.

REFERENCES

[1] Sheridan, T.B. (2002). Humans and automation: System design and research issues. New York: Wiley.

[2] Kirwan, B. and Flynn, M. (2002b). Towards a controller-based conflict resolution tool – a literature review. ASA.01.CORA.2.DEL04-A.LIT.

[3] Mercado Velasco, G.A., Mulder, M., & van Paassen, M.M. (2010). *Analysis of Air Traffic Controller Workload Reduction Based on the Solution Space for the Merging Task*. Proceedings of the AIAA Guidance, Navigation, and Control Conference, Toronto, Canada.

[4] Christoffersen, K., & Woods, D. D. (2002). How to make automated systems team players. *Advances in Human Performance and Cognitive Engineering Research*, 2, 1-1.