Design of human-machine teams using a modified CoActive Design Method

Professor Chris Baber[1], Chris Vance[2]

[1]University of Birmingham, [2]MBDA

ABSTRACT

Designing Human-Machine Teams not only requires an appreciation of which functions might be appropriately allocated to human or machine, but also how each team member can make sense of the functions performed by it and its team-mates. The aim of this paper is to present an approach to Allocation of Function within Human-Machine Teams (HuMaT) that can be applied across different Levels of Automation and which can explore information management issues in such teams. To do this, we present a modification of the CoActive Design method. A key aspect of the modification lies in the focus on information exchange and issues relating to common ground in HuMaT. In this paper, Cognitive Work Analysis is used as the basis for the CoActive Design Method to explore how different Levels of Automation can be conceptualised. The benefit of such an approach is that is provides a decomposition of functions such that it is possible to see how, even in systems that have high-levels of autonomy, there remains a role for human operators. Taking the example of an in-car navigation system, we illustrate how each member uses information to support the functions allocated to them, and how common ground develops in the team.

KEYWORDS

Human-Machine Teams, CoActive Design, Levels of Automation.

Introduction

Human-machine teams involve combinations of people and technology performing functions in pursuit of a shared objective. This raises challenges of how we might allocate functions between humans and machines, and how to differentiate a human-machine *team* from other forms of human-machine *interaction*. As JCN 1/18 (2018) points out, *"…approaches which adopt the 'automate what you can, leave the humans to fill in the remainder' view are likely to build systems that are cheap, but less resilient or effective."* This echoes Bainbridge's (1990) well known assertion concerning the 'irony of automation'. This means that following the simple rubric of HABA-MABA ('Humans-are-better-at… / Machines-are-better-at') can result in disjointed and dysfunctional design. Thus, approaches have developed to focus on the suggestion that 'Automation' can take one of several levels (Sheridan and Verplank, 1978).

The Levels of Automation (table 1) approach provides a framework for the relationship between Humans and Machines. However, the approach is not without its critics. For example, the discrete levels might be too coarse to capture the nuances of system performance (Feigh and Prichett, 2014), particularly in highly dynamic situations. There is also the implication (inherited, perhaps, from HABA-MABA) that 'functions' can be swapped between Human or Automation system

components with little or no cost (Bradshaw et al., 2013). Finally, the idea of discrete 'levels' could miss the objective of creating interdependent systems (Dekker and Woods, 2002).

Table 1: Levels of Automation

Level	**Description**	
	Decision Options	**Actions**
1	human makes all decisions	Human performs all actions
2	Automation offers complete set of options	Human performs all actions
3	Automation offers selection of options	Human performs all actions
4	Automation suggests one decision option	Human performs all actions
5	Automation suggests one decision option	Automation performs the action if human approves.
6	Automation suggests one decision option	Automation allows the human a restricted time to veto before action.
7	Automation suggests one decision option	Automation performs action, then informs human.
8	Automation suggests one decision option	Automation performs action and informs human only if asked.
9	Automation suggests one decision option	Automation performs action and informs human only if automation decides.
10	Automation decides everything.	Automation performs all actions.

Against these criticisms, Kaber (2018) suggests that the Levels of Automation approach could be improved to provide *"…engineering models that can be…used as bases for predicting human and system performance, as well as operator workload and system awareness outcomes, to support automation design and implementation…"* (p.17). One way of developing 'engineering models' of Levels of Automation is in terms of information processing demands. Thus, for Kaber and Endsley (2004) system operation could be described in terms of the following stages:

- Monitoring (system states); Generating (strategy options); Selecting (best strategy option); Implementing (chosen strategy)

Similarly, Wickens et al. (1998) suggested that functions could be considered in terms of:

- Information acquisition; Information analysis; Action selection; Action implementation.

From this, one could decide whether Automation could be used for Information acquisition / analysis to support human decision making, or whether it should work autonomously for Action selection / implementation. While the concept of Levels of Automation can aid Allocation of Function decisions, it does not consider the ways in which Humans and Automation use and share information (although in their original formulation, this *was* an issue that Sheridan and Verplank, 1978, discussed). An essential aspect of Human-Machine Teams involves the management of Common Ground of the information that Human or Automation team members use. This can highlight when information should be translated, e.g., interpreted or presented across different formats. Even when information can be deemed to be in common, there is still the need to appreciate how the different agents could make sense of this information; in other words, to decide what goal is being pursued by an agent when it has a specific piece of information.

The CoActive Design Method

The CoActive Design Method of Johnson et al. (2014) begins with a specification of functions that need to be performed. The original approach employed Cognitive Task Analysis. In this paper, Cognitive Work Analysis (CWA) is used (following the suggestion of Burns, 2018). CWA allows consideration of the 'system' (rather than a focus on a dyadic interaction). In particular, this paper uses the Abstraction Hierarchy from Work Domain Analysis. Specifically, the analysis is applied to Purpose-related and Object-related Functions. Once functions have been defined, the next step is to identify the capacities and capabilities a team member needs to be able to perform that function. These capacities and capabilities are defined in terms of whether a team member (a 'Performer') might be able to perform a function unaided, might require support, or might not be able to perform the function. Additionally, another team member (a 'Supporter') might be able to offer support for the function, so that it can be performed more effectively or reliably (table 2).

Table 2: Capacity and Capability to Perform or Support Functions [from Johnson et al., 2018] The cells are colour-coded (green, yellow, orange, red) and we have added diagonal shading for yellow cells and horizontal shading for orange cells, in case the paper is reproduced in greyscale.

Team Member Role Alternatives	
Performer	**Supporter**
I can perform the function without help	My support can improve efficiency
I can perform the function but my reliability is < 100%	My support can improve reliability
I can perform some aspects of this function but need support	My support is required
I cannot perform this function	I cannot provide support for this function

From the combinations of Performer and Supporter in table 2, plausible team configurations can be defined for different levels of interdependence. This could be used to show which functions can be performed unaided (independent), or which functions can be supported by team members, as shown in table 3. Functions can be 'constrained', e.g., by the available capacity and capability of team members, which could limit whether or not they could take on the function, or by the reliability with which the team member could handle the function ('brittle').

Table 3: Opportunities for team working [adapted from Johnson et al., 2018]

Team Member Role Alternatives		**Interpretation**
Performer	**Supporter**	
Achievable	**Constrained**	Independent operation by performer is viable, but support could improve efficiency
	Constrained	Independent operation by performer is viable, but support could improve reliability
	Independent	Independent operation by performer is necessary
Brittle	**Constrained**	Performer is <100% reliable, but support could improve efficiency

Constrained	**Constrained**	Performer is <100% reliable, but support could improve reliability
	Independent	Performer is <100% reliable, but no support is possible
	Constrained	Support can improve efficiency
	Constrained	Support can improve reliability
	Independent	Performer requires support, support can provide this
Unachievable?		Performer requires support, but none is possible
Unachievable		Function cannot be performed

Applying and modifying the CoActive Design Method

In this worked example, we use the simple scenario of using a Satellite Navigation (SatNav) system to support a driver in a car. In this example, Level of Automation 4 (i.e., Automation suggests one alternative and the human responds to that suggestion through their control actions, but is free to disregard this) is contrasted with Level of Automation 9 (i.e., Automation makes decisions and informs human only if automation decides to). The aim is to illustrate the benefit of CoActive Design Method for discussing Allocation of Function. We elaborate the method to provide a simple heuristic for scoring design options, and then, in the next section, we consider Information Requirements and common ground.

The first step is to define the primary functions of the system (in this case, we assume that the components of the system will be the car, the driver, the SatNav, the passenger). Thus, in the system there will be one Performer and up to three Supporters for each function. From CWA, the system's functions are defined in Table 4.

Table 4: Defining Functions for SatNav example

FUNCTIONS from Work Domain Analysis		
Abstract function	Purpose-related function	Object-related function
Correct destination	Confirm destination	define destination
		program destination
		follow route
		read signage
Best route	Confirm destination	define destination
		program destination
		follow route
		read signage
	Fuel efficient driving	follow route
		check fuel
	Fastest route	follow route
		avoid obstacles etc
Safe driving	Fastest route	follow route
		avoid obstacles etc

Based on the Object-related function column (in table 4), we can propose a Level of Automation 4 design in which the driver has primary responsibility but is assisted by automation (table 5). Table 5

uses the colour coding of functions (following table 2), with communication links, out (x) and in (⇐ or ⇒), between team members. In addition, colours have been arbitrarily scored on an ordinal scale (green = 3; yellow / diagonal stripe = 2; orange / horizontal stripe = 1; red = 0) to produce a 'total score' for each team member.

Table 5: LoA4: Driver in charge

From table 4		Performer	SupportA	SupportB	SupportC
Object-related function	Responsibility	DRIVER	PASSENGER	SATNAV	CAR
define destination	Human selects destination	x		⇒	
program destination	SatNav requires data	x		⇒	
follow route	SatNav defines routes	⇐		x	
read signage	Human can read signs	x			
define destination	Human selects destination	x		⇒	
program destination	SatNav requires data	x		⇒	
follow route	SatNav defines routes	⇐		x	⇒
read signage	Human can read signs	x			
follow route	SatNav defines routes	⇐		x	
check fuel	Human decides to refuel	⇐		x	x
follow route	SatNav defines routes	⇐		x	
avoid obstacles	Human steers car	⇐		x	
follow route	SatNav defines routes	⇐		x	
avoid obstacles	Human Steers car	⇐		x	
	Total Score	35		24	2
	Out	18 (0.51)		20 (0.83)	1 (0.5)
	In	17 (0.49)		4 (0.17)	1 (0.5)

The scoring (in table 5) has been applied to functions which are initiated by a team member (sum of 'out' scores) or which the team member responds to another instruction (sum of 'in' scores). Thus, for the Driver column, there are 6 cells with an x (indicating links out) and all of these are coloured green, scored 3. So, the links out score for the driver is 6*3 = 18. There are 8 cells with ⇐, indicating links in, of which 2 are green (scored 3), 4 are yellow (scored 2) and 1 is orange (scored 1). So, the links in score is (2*3) + (5*2) + (1*1) = 17. From this scoring, compared with the SatNav, the Driver has the higher score and the Driver in and out scores are similar. However, the SatNav has a much higher proportion of 'out' scores than the Driver. Thus, as you would expect, the Human (Driver) has more responsibility than the Automation (SatNav), but that many of the Driver functions required decisions performed in response to the SatNav.

Now, assume that the car is autonomous and cooperates with the SatNav (so does not require a Driver). This scenario is illustrated by table 6.

Table 6: LoA9: Car in Charge

From table 4		Performer	SupportA	SupportB	Support C
Object-related function	Responsibility	SATNAV	DRIVER	PASSENGER	CAR
define destination	SatNav offers destination	x		⇒	
program destination	Human confirms destination	⇐		x	
follow route	SatNav directs car	x			⇒
read signage	Car reads road signs; human confirms	⇐		x ⇐	x
define destination	SatNav offers destination	x		⇒	
Confirm destination	Human confirms	⇐		x	
follow route	SatNav directs car	x			⇒
read signage	Car reads road signs; human confirms	⇐		x⇐	x
follow route	SatNav directs car	x			⇒
check fuel	Car informs SatNav	x			⇒
follow route	SatNav directs car	⇐			x
avoid obstacles	Car informs SatNav	x			⇒
follow route	SatNav directs car	x			⇒
avoid obstacles	Car informs SatNav	x			⇒
	Total Score	42		10	19
	Out	27 (0.64)		6 (0.6)	5 (0.26)
	In	15 (0.36)		4 (0.4)	14 (0.74)

From table 6, one can see that the SatNav has the highest total score. The Human ('passenger') is unable to perform several functions (coloured red in table 6). However, notice that the Human is not passive (and has a similar proportion of 'out' scores to the SatNav). This means that the Human (in this design), even though 'out-of-loop' for several functions, is still initiating some of the functions and this involves making decisions. The design team could, on the basis of table 6, decide to reduce still further the opportunity for the human to act (and make the automation responsible for these functions) or could decide that the Human needs to be kept informed of the status of the 'out-of-loop' functions. In this design, the Car is responding to inputs from the other team members. In this example, the 'autonomy' of the Car is constrained by the actions of its team mates because the

activity concerns driving to a specific destination (rather than solely managing obstacle avoidance, lane handling and speed control that one might associate with 'autonomous' cars).

Defining Information Requirements and Potential for Common Ground

While the preceding examples rely on communication between team members, the method (as originally specified) does not tell us what is being communicated. To this end, we modify the Co-Design Method and create an additional table to indicate the information requirements of team members (table 7) and which highlights overlap of information use to consider the 'translation' required between team members and what common ground might exist in their collaboration.

Table 7: Information Requirements for SatNav example

From table 4	**Information requirements**		
Object-related function	**Human**	**Common**	**Automation**
define destination	geographical / physical location	post-code	GPS coordinates
program destination	alphanumerics on keypad	address details	ASCII (or other format)
follow route	directions	car movement	GPS coordinates
read signage	text on signs	-	-
define destination	geographical / physical location	address format	GPS coordinates
program destination	alphanumerics on keypad	address format	ASCII (or other format)
follow route	directions	car movement	navigate by waypoint
read signage	text on signs	-	-
follow route	directions	car movement	navigate by waypoint
check fuel	arrow on gauge	fuel level	estimate fuel required
follow route	directions	car movement	navigate by waypoint
avoid obstacles etc	things on road	car movement	congestion reports
follow route	directions	car movement	navigate by waypoint
avoid obstacles etc	things on road	car movement	congestion reports

The first function in table 7 is 'define destination'. For the human, this would be the physical or geographical location that needs to be reached, perhaps relating this location to previous journeys, knowledge of the road network, or expectation of the type of building to look for. This knowledge is translated into a single data format, i.e., a post-code. For the SatNav, the 'knowledge' of an address involves looking up the post-code in a table of Global Positioning System (GPS) coordinates. From the post-code, the SatNav could offer options to select a street name and house number to home in on a specific point. For the SatNav, the concept of a location exists as a defined point in (GPS) space, and the planning of a route involves determining which roads to use from the current point in space to this location, perhaps optimising for road type or fuel economy. Modern SatNavs are also able to plan journeys taking into account constraints, such as predicted traffic levels. Of course, this does not prevent SatNavs providing erroneous advice, with people who lack sufficient geographic knowledge unable to challenge this (MacKinlay, 2016). So, focusing solely on the alphanumeric description of location (without considering geographic aspects) could result in

poor understanding by the human and low common ground between human and automation, particularly in terms of issues such as 'passability' for large goods vehicles.

Several of the functions in table 7 have 'car movement' as common information. This implies that the 'human' has little ability to anticipate the activity of the car and so, is unable to intervene, until the steering manoeuvre has been made. This is not the place to discuss the relative merits of human versus autonomous driving, but does indicate an interesting challenge for HuMaT: should our design focus on allowing the human to intervene and resume 'control'? Or should the design focus on defining the higher-level 'policy' under which the automation operates? If the latter is acceptable, then a display could be provided to show that the 'policy' is being adhered to and to show the likely position of the vehicle in the near-future (so the human is able to monitor what the automation intends to do and why it intends this).

Conclusions

While the Levels of Automation approach is not without its critics, there continues to be interest in approaches that allow design decisions to be explored in systematic (if very qualitative) ways. In this paper, we have modified and extended the CoActive Design Method method through the consideration of information needs and their relation to common ground in HuMaT. We would anticipate this method being used as a support for sketching out alternative configurations of HuMaT in order to focus design decisions on how different functions could be performed and what information will be required. We also note that (considering the 'scores' in tables 5 and 6) one can use the method to consider the relative benefit of adding automation; that is, if the 'in' or 'out' scores obtained from a LoA 6 description of the task do not differ from those obtained from levels 7 to 9, then there is little benefit from adopting the higher levels. We also believe that the activity of relating the functions to human or automation gets the design team to consider how (or even why) a given function needs to be automated (and, conversely, to indicate the cost or benefit of leaving a function to be performed by a human).

Consideration of information-in-common (in table 7) provides an opportunity to consider how different members of a team might use the information available to them, and how such information might need to be translated in order to support effective communication. In this respect, the modifications that we have made help to shift the focus of the method from human-automation *interaction* towards consideration of human-machine teams (HuMaT). As noted in the Introduction, a defining feature of human-machine teams is the manner in which common ground (in information terms) is managed. When it is possible to define information in common, then one could envisage a user interface that presents this information to the human. However when the information, that is used by the human and the automation to perform the same function, differs then there is scope for misunderstanding or miscommunication. In these circumstances, the design team might consider how best to present the information to the human. For table 7, we noted that many of the items in the 'common' column related to the movement of the car. Rather than providing a means by which the human can intervene in response to such information, it might be more beneficial to provide a display that shows (in advance) when steering manoeuvres will be automatically performed and to show how such actions relate to the policy agreed between human and automation. If the actions become unacceptable (for example, the car is swerving wildly to avoid many potholes in the road, when it might be as well to slow down and drive over them), then to human might wish to alter the policy.

This is, of course, a pen-and-paper exercise but nevertheless, provides an opportunity for the design team to raise and explore concerns relating to the combination of humans and automation into teams and the potential gains to be obtained from using different levels of automation. While we have called this a 'design' method, its utility lies in the simple way in which it can represent the allocation of function and the sharing of information during the conceptual design stage. We argue that this approach could contribute to Early Human Factors Analysis, and could be useful for thinking the new sets of demands that HuMaT will raise for Ergonomics.

Acknowledgements

This work has been performed in partnership with the Defence Science and Technology Laboratory (Dstl) and the Direction Générale de l'Armement (DGA), and funded under the Materials and Components for Missiles, Innovation and Technology Partnership (MCM ITP).

References

Bainbridge, L. (1983) Ironies of automation, *Automatica 19*, 775–779.

Bradshaw, J. M., Hoffman, R. R., Woods, D. D., and Johnson, M. (2013) The seven deadly myths of "autonomous" systems, *IEEE Intelligent Systems, 28*, 54–61.

Burns, C.M. (2018) Automation and the Human Factors race to catch up, *Journal of Cognitive Engineering and Decision Making, 12*, 83-85.

Dekker, S. W. A., & Woods, D. D. (2002) MABA-MABA or abracadabra? Progress on human-automation co-ordination, *Cognition, Technology, and Work, 4*, 240–244.

Feigh, K. M., & Pritchett, A. R. (2014) Requirements for effective function allocation: A critical review, *Journal of Cognitive Engineering & Decision Making, 8,* 23–32

JCN1/18 (2018) *Human-Machine Teaming*, Ministry of Defence Joint Concept Note 1/18

Johnson, M., Bradshaw, J.M., Feltovich, P.J., Jonker, C.M., van Riemsdijk, M.B. and Sierhuis, M. (2018) CoActive design: designing support for interdependence in joint activity, *Journal of Human-Robot Interaction, 3,* 43-69.

Kaber, D. B. (2018) Issues in Human-Automation Interaction Modeling: presumptive aspects of frameworks of types and levels of automation, *Journal of Cognitive Engineering and Decision Making,12,* 1-18

Kaber, D. B., and Endsley, M. R. (2004) The effects of level of automation and adaptive automation on human performance, situation awareness and workload in a dynamic control task, *Theoretical Issues in Ergonomics Science, 5,* 113–153.

MacKinlay, R. (2016) Technology: use or lose our navigation skills, *Nature,* 573-575.

Sheridan, T. B., and Verplank, W. L. (1978) *Human and computer control of undersea teleoperators* (Technical report), Cambridge, MA: MIT, Man Machine Systems Laboratory.

Wickens, C. D., Mavor, A. S., Parasuraman, R., and McGee, P. (1998) Airspace system integration: The concept of free flight, In C. D. Wickens, A. S. Mavor, & J. P. McGee (Eds.), *The Future of Air Traffic Control: Human Operators and Automation*, Washington, DC: National Academy, 225–245.