

HUMAN-AUTONOMY TEAMING - AN EVOLVING INTERACTION PARADIGM: A COGNITIVE ENGINEERING APPROACH TO HAT

Axel Schulte and Diana Donath
Bundeswehr University Munich, Germany

This contribution outlines a cognitive engineering approach to structure, describe and depict configurations for highly automated human-machine systems using a common language. These systems involve cognitive agents, for autonomous vehicle guidance and mission management. The method focusses on the systematic top-down deduction of requirements for human-autonomy work share and interaction in the work process. Therefore, this contribution outlines a procedure to follow to design and describe such human-autonomy teaming systems, related user and system requirements, and top-level system designs. This contribution primarily aims at the application field of military, highly automated manned/unmanned vehicle systems.

Today, higher cognitive functions (e.g., perception, planning, and decision-making), traditionally exclusively owned by the human, are becoming an integral part of automated functions. In the last one or two decades the term “autonomous system” has widely been used to describe complex automated systems working largely independent from a human operator. However, the more capable the automation becomes, the more essential the issue of human-system functional allocation and integration has turned out to be (Klein et al., 2004). We share the concern of Bradshaw et al. (2013) that an undifferentiated use of the term of “autonomy” and the proliferation of automation can lead to unfruitful discussions and oddly defined development programs. We see the need for a conceptual framework unifying the nomenclature and description of systems in which human operators interact with complex automation.

Therefore, in this paper (the second part of the panel topic), we attempt to identify and formally describe common grounds among researchers and practitioners in this field. Despite our concerns, we want to adhere to the term of Human-Autonomy Teaming (HAT) to describe systems in which humans work with highly automated agents. Where those agents carry attributes like “autonomous” or “intelligent”, we will assign the unified term *Cognitive Agent*. Our approach, in general, suggests a *common semantical and graphical language* (Schulte & Donath, 2018), as well as a procedure to follow, to describe systems, system requirements, and top-level system designs. Both have a stronger focus on human-automation work share and integration aspects than traditional systems engineering practices and tools (e.g., Unified Modelling Language, UML). The traditional approach focuses solely on the formulation of requirements and the design of the technical functions of a system. The human operator only appears as an actor, usually located outside the system boundary. This approach is reasonable when automation is relatively simple, in the sense that it can perform specific clear-cut part-tasks. There, one can well describe the relationship between the (technical) system and the human user through use cases calling for certain user-system supervisory control interactions.

In this paper, in contrast, we want to take account of the following trends. Firstly, the automation in HAT will become much more capable in the sense of being able to perform higher cognitive tasks. Secondly, the work share and interaction between the user and the automated system will be much less stable (e.g. adaptive automation; as in Scerbo, 2006). Finally, the task

performance of human and automation will be highly dependent on a cognitive level (Hollnagel & Woods, 2005).

Building Blocks of a Language Describing HAT-Systems

Hence, we follow a strict systems engineering top-down approach to establish a formal description language, semantical and graphical, for highly automated human-machine and HAT-systems. In a first step, we introduce the notion of the Work Process (WProc), and its graphical representation (cf. Fig. 1), to develop an integrated view upon the purpose of a human-machine co-action, its physical and conceptional work environment (WEnv), as well as its desired output (WPOut) to the environment the WProc lives in. However, most crucial is the Work Objective (WObj), i.e., the mission or the purpose of work, since it reflects the user requirements for a system we will design. The WObj defines and initiates the WProc. The proper definition of the WObj is of high priority and most critical for the definition of the system boundaries and the design. Connections and dependencies between multiple WProcs (e.g., hierarchical or networked structures) we also capture and describe on this level. For a more detailed discussion, please refer to [6][7].

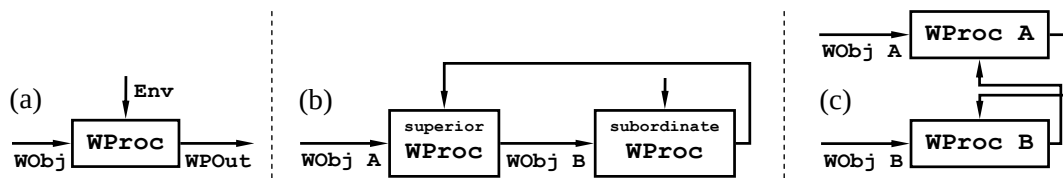


Fig. 1. (a) Work Process; (b) hierarchical; (c) networked.

In the second step, we establish a physical system instantiating the considered WProc. We call this system a Work System (WSys) in which the elementary roles of Worker and Tools are usually taken by humans and machines including conventional automation. The main characteristic of the role of the Worker is *to* know, understand, and pursue the WObj *by own initiative*. Without this initiative, the WProc would not be carried out. In principle, the Worker, and only the Worker, might as well self-assign a WObj. The Tools, on the other hand, will receive tasks from the Worker and will only perform them when told to do so. Hence, the Worker and the Tools are always in a Hierarchical Relationship (HiR, see green connector in Fig. 2, b). Again, for further reading, we recommend Schulte et al (2016) and Schulte & Donath (in prep).

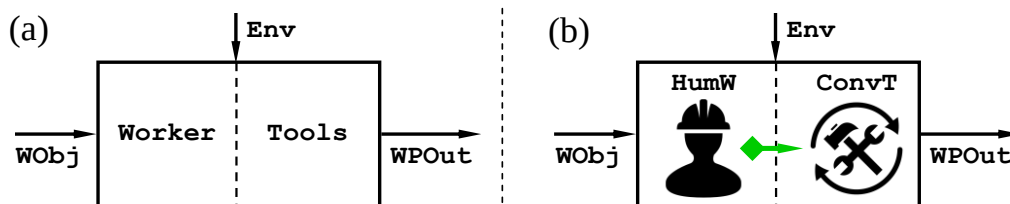


Fig. 2. Work System as physical instance of the corresponding WProc: (a) comprising the roles of the Worker and the Tools; (b) instantiated with a Human Worker (HumW) and Conventional Tools (ConvT).

Introduction of the Cognitive Agent into the Work System

In the third step, we introduce “the autonomy” into the WSys, represented by one or more Cognitive Agent(s) (CogA, little ‘R2D2’ in Fig. 3) in various roles and relationships to the human operator(s), the conventional automation, and the machinery.

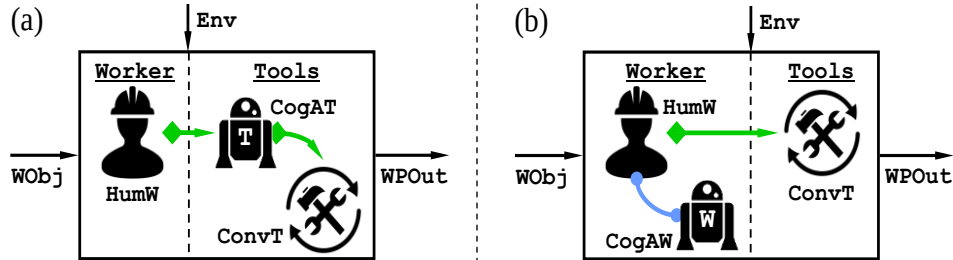


Fig. 3. (a) WSys with CogA as Tool (i.e., CogAT) in HiR;
(b) WSys with CogA as Worker (i.e., CogAW) in HeR.

Two trends have been followed in the past two decades concerning the role such a CogA could potentially take in system design. Firstly, so-called autonomous systems, i.e. systems that aim at performing user-given tasks, as much as possible independent from human intervention (see Fig. 3 (a): here the CogA works as Tool in a HiR supervised by a HumW). From a human factors stance, this design pattern will mostly serve the design goals to increase the human’s effectiveness,

to increase the human’s span of control, to reduce the human’s taskload, and others. Secondly, decision support, assistant, or associate systems, acknowledging that a human predominantly performs the work, while supported by a machine agent (see Fig. 3 (b): here the CogA is part of the Worker). In the latter case, there exists a Heterarchical Relationship (HeR, blue connector) between the HumW and the CogA being part of the Worker here. From a human factors stance, this design pattern will mostly serve the design goals to avoid or correct human erroneous action, to moderate or modulate human mental workload, to increase the human’s situation awareness, and others.

To sum up at this stage, the design options in the different applications are plenty (Schulte & Donath, in prep), still constructed, however, of only a few elementary building blocks. In our language, we represent those building blocks by a handful of the aforementioned semantical and graphical descriptors. The descriptors stand for actors (i.e., humans, cognitive agents, conventional automation including machines), role allocations of actors (i.e., worker, tools), relationships amongst actors (i.e., hierarchical, heterarchical), and a few others of less importance (e.g., co-location, grouping), not explicitly mentioned here.

Actor-Relationship-Actor Tuples as Master Design Patterns

From the various application-specific WSys designs using those descriptors in a coherent manner, we can now identify similar elementary (actor-relationship-actor)-tuples, as we call them. Thereby, we repeatedly can identify structural similarities in the sense of master design patterns for HAT, over a wide range of extremely different applications and system design approaches. With this, scientists and practitioners can start identifying common or alternative solutions for similar problems, namely design patterns for HAT-, and conventional human-machine systems.

Fig. 4 gives an annotated overview of all possible tuples. Here, the possible actors are HumW, CogA, and ConvT; possible relationships are HiR, and HeR. The HumW will, per definition, always take the role of a Worker; the ConvT will always be Tool, whereas the CogA may take the role of a Worker (i.e., CogAW), or of a Tool (i.e., CogAT). Between a Worker and a Tool there will always be a HiR.

	Worker		Tools	
(a)				
	HumW-HumW HiR	✗	✗	✗
	HumW-CogAW HiR	CogAW-CogAW HiR	✗	✗
	HumW-CogAT HiR	CogAW-CogAT HiR	CogAT-CogAT HiR	✗
	HumW-ConvT HiR	CogAW-ConvT HiR	CogAT-ConvT HiR	ConvT-ConvT HiR
	Worker		Tools	
(b)				
	HumW-HumW HeR	HumW-CogAW* HeR	✗	✗
	HumW-CogAW* HeR	CogAW-CogAW HeR	✗	✗
	✗	✗	CogAT-CogAT HeR	✗
	✗	✗	✗	ConvT-ConvT HeR

Fig. 4. (a) Hierarchical, (b) heterarchical (actor-relationship-actor)-tuples; (shaded: human involved; *: equal configurations; cross: invalid option).

A hierarchy of a CogA or a ConvT over a HumW, or a ConvT over a CogA is not reasonable. The same applies to a heterarchy of a ConvT with either a HumW or a CogA, for obvious reasons. Tuples, which do not involve humans, may not directly be interesting for HAT-systems. However, they certainly can influence the behavior of the automation “under the hood”, and therefore, be worthwhile to look at, at least from a pure engineering stance. Also, the pure human-human relationships, either hierarchical or heterarchical, may not directly be relevant for HAT-systems, except of course for WSys with more than one human. Apart from that, they may serve as valuable source for design metaphors. Finally, we do not want to allow a Human-Agent HeR, where the agent is part of the Tools, since per definition there is always a HiR between Worker and Tools.

Example: Manned-unmanned Teaming Helicopter Mission

As an example, we look at a WSys for a Manned-Unmanned Teaming (MUM-T) military helicopter mission. On the level of WProcs, there is also involved a command and control (C2) WProc that provides the MUM-T mission WObj. The Work Object (WO) is the troops to be transported in or out a combat zone. Fig. 5 depicts the WSys setup. It consists of a cockpit crew of two humans, pilot flying and commander, and a CogAW representing a crew associate system. The roles of the pilot flying and the commander the crewmembers may swap amongst each other at any time. The Tools are a manned transport helicopter (H/C), where the aforementioned Workers are located, and three dislocated small reconnaissance UAVs (Unmanned Aerial Vehicles), each of which controlled by an on-board CogAT that provides a delegation interface to the cockpit crew for highly automated UAV tasks. The crew associate may also directly use this interface and the high-level commands supported by the CogATs onboard the UAVs.

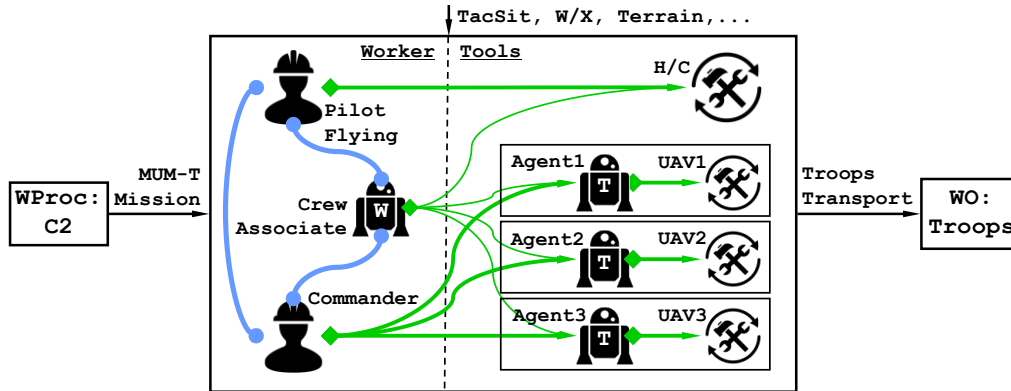


Fig. 5. WSys setup for a military Manned-Unmanned Teaming (MUM-T) mission.

As Fig. 5 indicates, there are a number of different (actor-relationship-actor)-tuples involved in the system setup. Particularly, the (Commander-HeR-Associate)-tuple is of high relevance for HAT considerations. Here, we implemented it by use of a mixed-initiative design pattern for mission planning tasks (Schmitt et al, 2018), and by a workload-adaptive design pattern for mission execution tasks (Brand et al, 2018), both optionally blended with the SA-based Agent Transparency (SAT) Model (Chen et al, in press). The (Commander-HiR-AgentX)-tuple is implemented by using our well-proven task-based guidance concept (Uhrmann & Schulte, 2012; Rudnick & Schulte, 2017). Addressing the UAVs as a team is also an implemented option.

Discussion

In this contribution, we briefly outlined a description language and procedure to follow for a systematic top-down approach for the definition of Human-Autonomy Teaming (HAT) systems. This approach tries to formalize the description of complex, highly automated human-machine systems, in particular, in the domain of manned and unmanned vehicle guidance and mission management applications. The resulting system representations allow the discussion of system characteristics on a common high level of abstraction, using only a few descriptors. Recurring structures of human-agent collaboration can also be identified easily. Additionally, the discovery, the discussion, and the exchange of beneficial design patterns for HAT are facilitated. Future works will aim at a further formalization of the language. Furthermore, we will need to strengthen the linkage between the system-level description of HAT-systems and the characterization of individual design patterns.

References

- Bradshaw, J.M., Hoffman, R.R., Woods, D.D., & Johnson, M. (2013) The Seven Deadly Myths of "Autonomous Systems". IEEE Intelligent Systems, (3), pp. 54-61.
- Brand, Y, Ebersoldt, M., Barber, D., Chen, J.Y.C., & Schulte, A. (2018) Design and Experimental Validation of Transparent Behavior for a Workload-adaptive Cognitive Agent. Intelligent Human Systems Integration (iHSI).

- Chen, J.Y.C., Lakhmani, S., Stowers, K., Selkowitz, A., Wright, J., & Barnes, M. (in press) Situation Awareness – Based Agent Transparency and Human-Autonomy Teaming Effectiveness. *Theoretical Issues in Ergonomics Science*.
- Hollnagel, E., Woods, D.D. (2005) *Joint cognitive systems: Foundations of cognitive systems engineering*. CRC Press.
- Klein, G., Woods, D.D., Bradshaw, J.M., Hoffman, R.R., & Feltovich, P.J. (2004) Ten challenges for making automation a “team player” in joint human-agent activity. In: *IEEE Intelligent Systems*, (6), pp. 91-95.
- Rudnick G., Schulte A. (2017) Implementation of a Responsive Human Automation Interaction Concept for Task-Based-Guidance Systems. *Lecture Notes in Computer Science*, Volume 10275. Springer-Link.
- Scerbo, M. (2006) Adaptive automation. In: *Neuroergonomics*. Pp. 239-252.
- Schmitt, F., Roth, G., Barber, D., Chen, J.Y.C., & Schulte, A. (2018) Experimental Validation of Pilot Situation Awareness Enhancement through Transparency Design of a Scalable Mixed-Initiative Mission Planner. *Intelligent Human Systems Integration (iHSI)*.
- Schulte, A., Donath, D., & Lange, D.S. (2016).: Design Patterns for Human-Cognitive Agent Teaming. *Lecture Notes in Computer Science*, Volume 9736. Springer Link.
- Schulte, A., & Donath, D. (in prep) Systems Level Design Patterns Approach for HAT. In: *Human-Autonomy Teaming: Supporting Dynamically Adjustable Collaboration*. NATO STO Task Group Report HFM-247. Neuilly-sur-Seine, France.
- Schulte, A., & Donath, D. (2018) A Design and Description Method for Human-Autonomy Teaming Systems. In: Karwowski W., Ahram T. (Eds). *Advances in Intelligent Systems and Computing*. Vol 722. Springer, Cham.
- Uhrmann, J., & Schulte, A. (2012) Concept, Design and Evaluation of Cognitive Task-based UAV Guidance. *IARIA Journals*. ISSN-Online 1942-2679.