A Groupware and Automation Design Perspective on HumanS-AutomationS Teaming

Philippe Palanque

ICS-IRIT, Université de Toulouse, 31042 Toulouse Cedex, France

Abstract

Human Machine Teaming or Human Automation Teaming has mainly considered one automation interacting with one operator. The field of Computer Supported Cooperative Work has been studying for decades how groups of users can collaborate to reach objectives out of reach of an individual. This position paper proposes a generic framework for designing and assessing collaboration between a group of users/operators and a set of interactive systems partly autonomous. This framework exploits frameworks both from automation research and CSCW. The paper embeds those frameworks to discuss potential issues in hybrid automation such as transparency, controllability and situation awareness.

Keywords

Human-Automation Teaming, Groupware, CSCW, Automation

1. Introduction

The field of Computer Supported Cooperative Work has been studying for more than forty years how humans may collaborate to carry out work. Recent work focus in this area has in last decades moved away from the notion of work to the notion of entertainment with the advent of social media and more globally social computing. In that context early work on human-human collaboration is relevant when studying how humans collaborate together.

This position paper presents two frameworks which have been widely used in the area of CSCW to better understand and characterize human-human collaborations. We also present the GUSPATO model which provides a scale of collaboration types between humans and automation. This framework makes it possible to define or assess how collaboration may take place and how automation may be intertwined with the collaboration. This paper builds on a previous paper [13] providing more details on the intersection between collaboration and automation. In the context of safety critical systems where various users with various qualifications working both synchronously and asynchronously (possibly with high time difference) these issues should be addressed as first class citizens as demonstrated for satellites collision avoidance systems [12] and satellites ground segments [14].

The rest of the position presents how these models are useful and can be applied to hybrid automations.

2. CSCW/Groupware frameworks

The description of human-human collaboration requires the explicit identification of the tasks that each human has to perform, in order to understand and allocate the work between them. It also requires the identification of specific aspects of the collaboration which will take place between users but also between users and the interactive system functionalities.

palanque@irit.fr (P. Palanque)
0000-0002-5381-971X (P. Palanque)

Cooperative work may be dedicated to one or more of the following types of collaborative activities: production, coordination, communication. To make these three facets more explicit, let's use the example of a collaborative text editing tool. In that context, a coordination tasks (and its associated tool) would allow users to coordinate their collaboration by making explicit which user is allowed to work on the document at what time of day, as well as making explicit the sequencing of these activities performed the users. A communication task (and its associated tool) would allow users to communicate by voice while working on the document. A production task (and its associated tool) would allow users to input text in the document. This means that collaborative application should provide functions triggerable by the users in order to collaborate effectively. It is then possible to associate one or more facet amongst this set. For example, Figure 1 a) shows that one task is dedicated to coordination whereas Figure 1 b) shows that the task is dedicated to both coordination and communication.

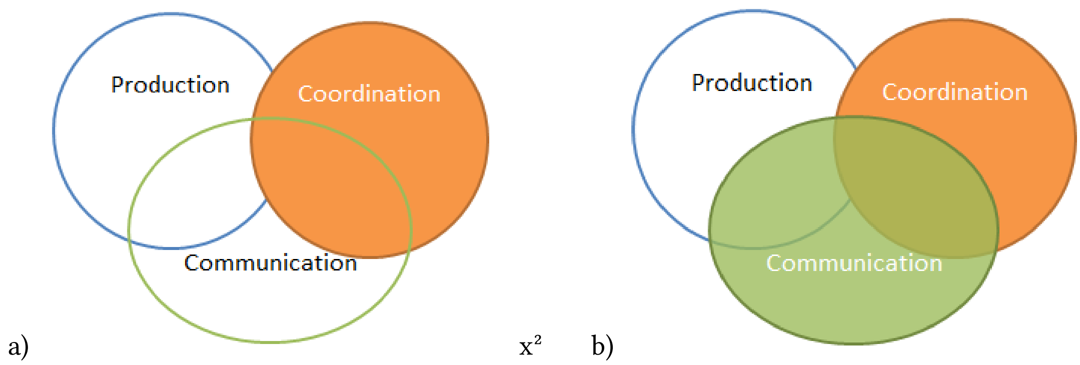

Figure 1. Example of collaborative tasks properties from a "functional clover" [10]

Collaborative tasks may be performed within various space-time constraints (local/distant, synchronous/asynchronous) [4]. Table 1 presents these different space-time constraints and illustrates the different possibilities with an example.

Table 1. Time Space Matrix from [6]

	Same Time	Different Times
Same Place	Face to face interaction Ex: two users editing the document in the same office. They could use concurrently the automation of a tape-recorder in order to check, on a later stage, the oral exchanges which took place in the room.	Asynchronous interaction Ex: one user edits the document during on a computer in the morning and another user edits the document in the afternoon. They could use the automatic formatting of text (based on styles) as an automation to support consistency in the formatting of the document. This automation will be used in sequence (when one user is editing the document).
Different Places	Synchronous distributed interaction Ex: Communication of one user in one office with another user in another office. They can use the automation of a video conferencing system to convey voice, image and the text being edited in a concurrent way.	Asynchronous distributed interaction Ex: two users are working at different times from different places on the same document. They could use a spellchecker embedded in the text editor as an automation to support the identification of spelling mistakes. This automation may be used in sequence (when each user is editing the document).

These frameworks are also useful when describing collaborations with hybrid groups (i.e. including automated agents and humans). For instance, the fact that

The functional clover framework can thus be integrated with the time/space matrix of Table 1. The Place and Time aspects will impact the three aspects of the functional clover adding their own perspectives on the work to be performed by the group of users.

As we address here collaboration with multiple entities, the basic mechanisms such as collaborative undo, semaphores, locking mechanisms, concurrent interactions will be needed in most of the possible configurations.

Table 2. Time Space Matrix and the Functional Clover

	Same Time	**Different Times**
Same Place	*Face to face interaction* *Communication*: Any tool or no tool fits *Coordination*: Any tool or no tool fits Production: Depends on the type of production, as production of digital content will need a dedicated tool. As multiple users are involved, the tool should support concurrent activities (and dedicated mechanisms such as collaborative undo or lock protections for access control).	*Asynchronous interaction* *Communication*: By definition only asynchronous communication in the same place. This can be implemented by shared boards. *Coordination*: Requires dedicated tool support to provide coordination means among entities that are not synchronously present and active. *Production*: Any tool or no tool would fit. In case of fully asynchronous not semaphores-based locking mechanisms are needed.
Different Places	*Synchronous distributed interaction* *Communication*: Remote synchronous communication mechanisms such as chat tools or conferencing systems. *Coordination*: The coordination tools depend heavily on the distribution of entities in various time zones. Tools should offer information about activities, presence, status … of each remote entity. *Production*: Tools for synchronous execution of production tasks.	*Asynchronous distributed interaction* *Communication*: As entities are fully remote multiple types of asynchronous communications need to be fully supported. *Coordination*: Coordination tools are required for supporting both asynchronous and distant coordination. *Production*: Production tools not necessarily require synchronous management of input and accesses, but versions and revisions management are critical as these problems cannot be solved by means of synchronous interactions.

The integration of Functional Clover and Time/Space matrix highlights the relevance of both aspects together when discussing and trying to understand collaborations. What is interesting is that the fact that a participating entity is a human or a software/system does not interfere with the description. Indeed, the tools and interaction means on these tools might differ but the concepts and their interactions remain. However, interactions between (partly-) autonomous entities bring specific challenges very well highlighted in the intrinsic behavior of humans as presented in the theory of mind principle [19]. In psychology, the theory of mind (ToM) refers to the inner functioning of a person when it attributes mental states to other people. These mental states of other people are then exploited by the person to predict/infer the behavior of the other people and thus to plan its own behavior [1].

People utilize a theory of mind when analyzing, judging, and inferring other people's behaviors. When interacting with partly-autonomous systems additional specific information about the behavior and the state of the entity should be shared with the humans so that they can apply their inner ToM behavior (as they do with other humans). **Figure 2** give a concrete example of ToM on the driving tasks for an autonomous versus a human-performed activity.

a)  b) 

Figure 2. An example of a) an autonomous car not supporting well the ToM behavior of humans (as no information about state and future behavior is shared) b) a human "sharing" her gaze so that people crossing the road can infer a lack of attention on the driving task.

3. The RCRAFT Framework for human-automation collaboration description

Whatever the design objectives are [17], the automation design requires the identification of specific elements which describe how this automation behaves. This section presents the RCRAFT conceptual framework, which provides a means to characterize the automation elements.

3.1. RCRAFT - Allocation of Functions and Tasks (FT)

The FT part of the RCRAFT conceptual framework deals with the issue of Allocation of Functions and Tasks. The generic term is usually function allocation [21] but RCRAFT differentiates the term function according to the entity that per-forms it. Activities performed by human operators are called "tasks" while the ones performed by the interactive system are called "functions". Automation designs will thus result in the identification of which activities are carried out by which entity (user or system or both). RCRAFT-based design of automation is thus based on the identification and the complete description of the human-system cooperative work as tasks and functions.

3.2. RCRAFT - Allocation of Resources (R)

The "R" part of RCRAFT deals with the identification and representation of re-sources used by the entities. "Resource" is a generic term used to cover data, information (in the head of the user), physical objects (e.g. a credit card), soft-ware objects (information manipulated by the system) and devices (input, output and input/output) required to perform the system functions and the human tasks. As for the "FT" part, automation designs will allocate resources to entities, which may or may not be shared it with other entities. In [15], the authors have shown that, in order to describe operators' tasks precisely, identification and representation of data related to those tasks are crucial. For instance, an automation not sharing information related to its state may result in automation surprises [18] on the users' side.

3.3. RCRAFT - Allocation of Authority (A)

The "A" part of RCRAFT is concerned with identifying the controlling entity (i.e. which entity can influence the situation so that it develops or continues in a way which satisfies its requirements [9]). The entity with authority performs or de-fines constraints on the tasks and functions that modify the state of a system, an object or the environment to reach a goal. These constraints, tasks or functions affect the resources needed by the user or the system. Automation designs may allocate authority globally (i.e. one entity is a considered as master and the other as slave [23]). This allocation may be static or dynamic/adaptive [22] i.e. changing over time to adapt to changes (context, workload ...).

3.4. RCRAFT - Allocation of Control Transition (CT)

The “C” part of RCRAFT deals with the issue of Control Transition, which de-fines how an entity may takeover control, hand it over or share it with another entity [20]. Automation designs will identify Control Transitions, which describe who can modify the allocation of control. As discussed above, it is possible that the entity that performs a task or a function that initiates a control transition may not have authority. An entity may release control without defining how control is to be allocated going forward, leaving that task to another entity with authority. This justifies the conceptual separation of the two entities.

3.5. RCRAFT - Allocation of Responsibility(R)

The second “R” in RCRAFT deals with the issue of Responsibility, which defines which entity can derive a specific outcome on which the user goal depends. This outcome is called the ‘result’. Automation designs will identify the list of expected results for the work, and will indicate which activity of which entity in-fluences one of the expected results of the work carried out jointly by the system and the user. The entity that influences one of the expected results will be said to be (at least partly) accountable for the outcome.

The RCRAFT framework introduced in [3] provides and comprehensive way of describing the relationships and the abstract behaviour of hybrid entities involved in a collaboration. More detailed low level descriptions are also needed but they will then correspond to a concrete solution to a given problem and this goes beyond this position paper.

4. The GUSPATO framework

The way humans can collaborate with automation is presented using the GUSPATO model in Figure 3. GUSPATO is the acronym composed with the first letter of the seven types of collaboration where control, authority and responsibility (according to the terminology of the RCRAFT framework for automation) migrate between the technical system embedding automation and the human.

4.1. Description of GUSPATO framework

Figure 3 presents a multi-layer model identifying the co-evolution of the collaboration between human and system as far as automation is concerned. Each line of the figure corresponds to one type of collaboration with automation. The first line describes a collaboration where the human is seen as a ‘god’ and creates the system and its outcome. In that case the system can be seen as an object belonging to the creator.

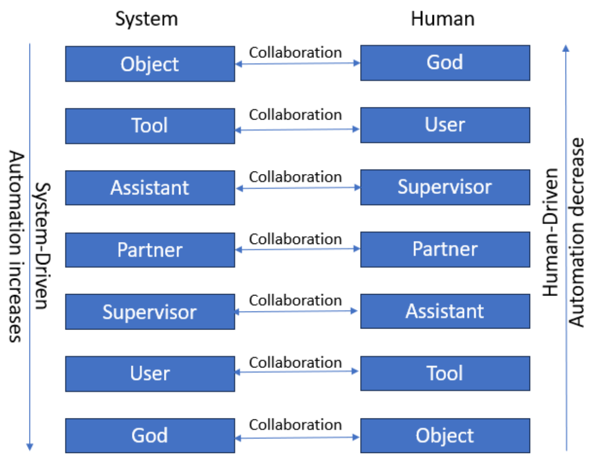

Figure 3. GUSPATO: A seven-level classification of Human-System Collaboration

The second line represents the classical use of computers where the system is seen as a tool used by the user or operator. The tool may embed some automation but the control, the authority and the responsibility remain with the user/operator. The human is, here, inside the interaction loop perceiving information provided by the system, cognitively processing it and triggering system functions when appropriate. A concrete example of this layer is the one of editing tools supporting the authors to produce an artefact such as a visual program [7].

Line three shows an unbalanced sharing of control, authority and responsibility between the system (as assistant) and the human (as supervisor). Automation is more complex and more complex tasks are performed by the system, following a delegation of tasks by the human. The human still holds control, authority and responsibility but positioned over the interaction loop (monitoring the partly autonomous behaviour of the system). A concrete example of this layer would be the addition of a spell checker in the editing tool mentioned above, as, for instance, in Microsoft Word text editor. In that case the spell checker allows the user to perceive potential spelling mistakes and decide or not to select one of the offered correction options (see Figure 4).

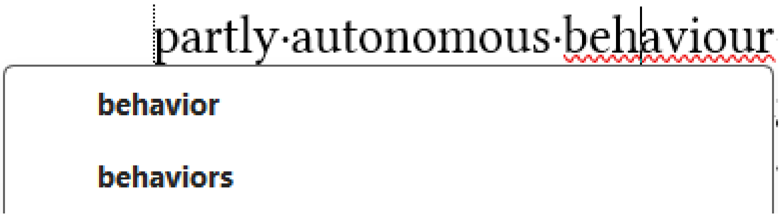

Figure 4. A concrete example of a spell checker level in MS Word text editor

Line four in the middle of the figure corresponds to a symmetric relationship for control, authority and responsibility for the collaboration between the system and the human. In this type of collaboration, both entities can delegate tasks to the other entity and monitor their performance. Authority and responsibility are shared, and the human can be considered as outside of the interaction loop when the systems perform tasks autonomously.

The fifth line corresponds a reversal of the collaboration presented in line three but now the human is an assistant to the system. In that context the system might require the human to perform tasks and will monitor the performance of the human. Such reversal of roles in the collaboration is similar for the last two lines of the figure. A concrete example can also be drawn from Microsoft Word text editor. If one types the text "hsi" (as in Human-System Integration) it is automatically changed to "his" by MS Word. The user cannot avoid this change (in a given configuration of MS word) and must edit again the text.

The last line corresponds, for instance, to generative AI where objects are created by the system where the human is considered as an object amongst many other ones. In that context users are mainly ignored and the system could be setup in a way that it generated an infinite number of images that no user will see.

4.2. Robust Automation in Safety-Critical Contexts

One key issue in the use of GUSPATO model is that the lower lines require more complex algorithms and, in some cases, might require the exploitation of AI technologies. While this might be acceptable for entertainment or mass-market systems like translation, complexity in computer systems is a precursor for failures (at least in the area of software where "Complexity metrics are better predictors than simple size metrics of fault and failure-prone" [8].

modules.). When new technologies for producing computing systems appear (a new programming language for instance) significant efforts are required to harden the technology making it suitable for deployment in critical contexts. This is the reason why it is wiser and safer to keep older technology in use and to refrain from being an early adopter in order to avoid disillusion, as is the case with the fantasy of fully autonomous driving [5].

5. Conclusion and Application Example

This position paper has presented different frameworks for the domain of CSCW and from the domain of automation design. These frameworks have demonstrated being useful when designing hybrid automations, especially in the large civil aircraft cockpits. For instance, the Flight Warning System (FWS) [4] and the Autopilot (AP) could be seen as two automations interaction jointly with the flying crew (i.e. the captain and the first officer). In the AP and the FWS automations are very different (in terms of level according to the GUSPATO model). According to the Ellis's framework for groupware, all the interactions take place in same place same time and the four entities provide function for cooperation, production and communication (according to the clover model). One of the key issues is to implement and verify such systems where collaborative and automated aspects add significant complexity to the software architecture and the code. As demonstrated in [2] formal methods and more precisely high-level Petri nets can support this difficult and often overlooked software engineering task while generic structuring mechanisms support maintainability and management of large quantities of code [16]

Acknowledgements

This work was partly funded by the project of the French Space Agency - CNES 21055/00.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] Baron-Cohen S. Precursors to a theory of mind: Understanding attention in others. In Whiten A (ed.). Natural theories of mind: evolution, development, and simulation of everyday mindreading. Oxford, UK Cambridge, Massachusetts: B. Blackwell. pp. 233–251 (1991).

[2] Bastide, R., & Palanque, P. (2001). Modeling a groupware editing tool with cooperative objects. In Concurrent Object-Oriented Programming and Petri Nets: Advances in Petri Nets (pp. 305-318). Berlin, Heidelberg: Springer Berlin Heidelberg.

[3] Bouzekri E., Martinie C., Palanque P., Atwood K., and Gris C. Should I Add Recommendations to My Warning System? The RCRAFT Framework Can Answer This and Other Questions About Supporting the Assessment of Automation Designs. In 18th IFIP TC 13 International Conference on Human-Computer Interaction – INTERACT. Springer-Verlag, Berlin, Heidelberg, 405–429 (2021).

[4] Bouzekri, E., Martinie, C., Palanque, P., Atwood, K., & Gris, C. (2021). Should I add recommendations to my warning system? The RCRAFT framework can answer this and other questions about supporting the assessment of automation designs. In Human-Computer Interaction–INTERACT 2021: 18th IFIP TC 13 International Conference, 2021, Proceedings, Part IV 18 (pp. 405-429). Springer International Publishing.

[5] Cusumano, M. Self-driving vehicle technology: progress and promises. Communication of the ACM 63, 10, 20–22, (2020).

[6] Ellis, C. A., Gibbs, S. J., & Rein, G. (1991). Groupware: Some issues and experiences. Communications of the ACM, 34(1), 39–58, (1991).

[7] Esteban, O., Chatty, S., Palanque, P. (1995). Whizz'ed: A Visual Environment for Building Highly Interactive Software. In: Nordby, K., Helmersen, P., Gilmore, D.J., Arnesen, S.A. (eds) Human—Computer Interaction. IFIP Advances in Information and Communication Technology. Springer, Boston, MA.

[8] Fenton N. E. and Ohlsson N. "Quantitative analysis of faults and failures in a complex software system," in IEEE Transactions on Software Engineering, vol. 26, no. 8, pp. 797-814, Aug. (2000).

[9] Flemisch, F., Heesen, M., Hesse, T., Kelsch, J., Schieben, A., Beller, J.: Towards a dy-namic balance between humans and automation: authority, ability, responsibility and control in shared and cooperative control situations. Cogn Tech Work. 14, 3–18. https://doi.org/10.1007/s10111-011-0191-6 (2012).
[10] Laurillau Y. and Nigay L. Clover architecture for groupware. In Proceedings of the 2002 ACM conference on Computer supported cooperative work (CSCW '02). Association for Computing Machinery, New York, NY, USA, 236–245 (2002).
[11] Martinie C., Bouzekri E., Palanque P. From Human-Human Computer Mediated Communication to Human-Automation Collaboration in the light of Large Civil Aircraft Workplace. Workshop on Automation Experience at the Workplace, AutomationXP 2021, co-located with the ACM Conference on Human Factors in Computing Systems (CHI 2021), May 2021, Yokohama, Japan. ⟨hal-03376276⟩
[12] Martinie, C., Barboni, E., Navarre, D., Palanque, P., Fahssi, R., Poupart, E., & Cubero-Castan, E. (2014, June). Multi-models-based engineering of collaborative systems: application to collision avoidance operations for spacecraft. In Proceedings of the 2014 ACM SIGCHI symposium on Engineering interactive computing systems (pp. 85-94).
[13] Martinie, C., Bouzekri, E., & Palanque, P. (2021). From Human-Human Computer Mediated Communication to Human-Automation Collaboration in the light of Large Civil Aircraft Workplace. In Workshop on Automation Experience at the Workplace, AutomationXP 2021, co-located with the ACM Conference on Human Factors in Computing Systems (CHI 2021) (Vol. 2905). CEUR Workshop Proceedings. xx
[14] Martinie, C., Navarre, D., & Palanque, P. (2014). A multi-formalism approach for model-based dynamic distribution of user interfaces of critical interactive systems. International journal of human-computer studies, 72(1), 77-99.
[15] Martinie, C., Palanque, P., Ragosta, M., and Fahssi, R.: Extending procedural task models by systematic explicit integration of objects, knowledge and information. 31st European Conference on Cognitive Ergonomics (ECCE '13). ACM, Article 23, 1–10 (2013).
[16] Navarre, D., Palanque, P., Bastide, R., & Sy, O. (2000, June). Structuring interactive systems specifications for executability and prototypability. In International Workshop on Design, Specification, and Verification of Interactive Systems (pp. 97-119). Berlin, Heidelberg: Springer Berlin Heidelberg.
[17] Palanque P.: Ten Objectives and Ten Rules for Designing Automations in Interaction Techniques, User Interfaces and Interactive Systems. In Proceedings of the International Conference on Advanced Visual Interfaces (AVI '20). ACM, Article 2, 1–10 (2020).
[18] Palmer, E.: Oops, it didn't arm'- A case study of two automation surprises. In International Symposium on Aviation Psychology, 8th, Columbus, OH (pp. 227-232) (1995).
[19] Premack D, Woodruff G. Does the chimpanzee have a theory of mind? Behavioral and Brain Sciences. 1 (4): 515–526 (1978).
[20] Tan, D., Chen, W., Wang, H., Gao, Z.: Shared control for lane departure prevention based on the safe envelope of steering wheel angle, Control Engineering Practice, Volume 64, Pages 15-26 (2017).
[21] Wright, P., Fields, R., Harrison, M. Analyzing Human-Computer Interaction as Distributed Cognition: The Resources Model. Hum. Comput. Interact. 15(1): 1-41 (2000).
[22] Wu, Y., Wei, H., Chen, X., Xu, J., Rahul, S.: Adaptive Authority Allocation of Human-Automation Shared Control for Autonomous Vehicle. Int. J Automot. Technol. 21, 541–553 (2020).
[23] Zhang, Z., Zhao, D.: Master-slave control strategy of tele-manipulator. International conference on Robotics and biomimetics (ROBIO'09). IEEE Press, 2063–2067 (2009)