Human-Autonomy Teaming Patterns in the Command and Control of Teams of Autonomous Systems

Douglas S. Lange(✉) and Robert S. Gutzwiller

Space and Naval Warfare Systems Center Pacific (SPAWAR),
San Diego, CA, USA
{doug.lange, robert.s.gutzwillerl}@navy.mil

Abstract. Design patterns have been found useful in several domains. This paper helps motivate their use in the field of human-autonomy teaming and provides three example patterns that could be contributed to the language of patterns available to system developers. In our examples we focus on the motivations and consequences in terms of human and team performance when describing the features of the individual patterns. These replace forces more commonly used in software engineering or other fields. Practitioners and researchers alike will benefit from a vetted vocabulary of established patterns of the form presented.

Keywords: Autonomous system · Cognitive agent · Cognitive assistant · Command and control · Design patterns · Teaming · Supervisory control · Systems engineering

1 Introduction

Command and control (C^2) has long been an example of human-autonomy teaming. In particular, commanders have been directing teams of autonomous systems since the concept of C^2 began. These autonomous systems were collections of infantry or artillery or perhaps squadrons of aircraft or ships. The decisions behind the autonomy being displayed came from biological brains rather than biologically inspired artificial intelligence.

1.1 Command and Control

Commanders guide the operation of a team. There is a goal or mission to be accomplished, an environment to operate in that may include an opposing force, and there are the team members themselves. In some cases the commander is separate from the team

This work was prepared by an employee of the United States government as part of the employee's official duties and is not subject to United States copyright. Foreign copyrights may apply.

D. Harris (Ed.): EPCE 2016, LNAI 9736, pp. 179–188, 2016.
DOI: 10.1007/978-3-319-40030-3_19

and in essence supervises the operation. In others, the commander is an integral part of the team (e.g., an infantry squad) and teams with the other autonomous units (i.e., the other members of the squad).

According to Willard [1], C^2 is accomplished through the following tasks:

- Ensure that all decisions remain aligned with the mission and the commander's intent.
- Assess the status of plan execution utilizing a common operational picture that is also provided to the team members.
- Monitor the status of the plan against the plan's timeline.
- Oversee compliance with procedures to avoid mistakes and achieve efficiencies.
- Respond to emerging information that differs significantly from expectations.
- Reapportion assets due to changes in availability or changes to requirements and priorities.

Beyond the control of autonomous teams or units, human-autonomy teaming has found its way into the command environment itself. Projects have shown that intelligent agents can assist a human commander in exercising C^2 [2]. Thus, human-autonomy teams can become responsible for controlling teams of autonomous systems employed to execute the mission.

SKIPAL [2] is an example where the role of autonomy is strictly within the task of maintaining situational awareness. The human's role is both as a consumer of the information and as an instructor to the autonomy. Other research [3–5] includes the possibility of utilizing autonomy to plan, monitor, and reapportion assets for missions. Autonomy can even respond on the human's behalf with permission of the human to handle rapidly occurring events [5].

1.2 Design Patterns

The use of a semi-formal language to describe design patterns began with the fields of architecture and land use [6, 7]. Christopher Alexander proposed that buildings and towns are created as collections of patterns that are the result of forces and processes. Communicating these patterns among practitioners in the field provides a powerful design tool.

Design patterns and pattern languages became a popular tool for software engineering in the 1990s and later for multi-agent systems [8–13]. It is proposed in [14] that the study of human-autonomy teaming adopt this approach for describing and communicating critical design patterns, enabling reuse of these patterns across systems and mission domains. To accomplish this goal, the same critical elements employed by previous pattern users must be described:

- Pattern Name
- The forces driving the problem to be solved
- The solution
- The positive and negative consequences of using the pattern
- Implementation advice

References [7–13] provide catalogs of some discovered patterns in their particular domains.

To begin the process, we must first provide a good if informal definition of the problem to be solved and the forces in play. A systems engineering view indicates that the work objective is the place to start [14]. This sets the goal for a work process that will be employed by a human-autonomy team. An environment must also be defined along with a system boundary. The objective and environment together form the forces for the problem. The process indicates the solution. In any process there will consequences, and in particular for a human-autonomy teaming process, there will be consequences on the performance of the human, the autonomy, and the joint human-autonomy team in how the process is defined. Different methods of implementation will affect how the consequences play out and the development of the capabilities desired.

In this paper, we attempt to describe patterns of human-autonomy teaming utilized in our development of task management aids for a human-autonomy team exercising the C^2 of teams of autonomous systems. These are not patterns that we ourselves first discovered. They have been experimented with by many others, and we know of them through research literature. To engage in dialog on human-autonomy teaming, we will attempt to employ the pattern methods suggested in [14].

1.3 Forces in Human-Autonomy Teaming

The forces leading to the motivation to utilize a particular software engineering pattern might include the effects of computational complexity or difficulty in extending the software for future requirements [8]. We propose to look at forces from human performance and teaming research [17, 18]. In particular, we will use the Input, Process, Output (IPO) model to derive forces. Examples of forces derived from this model are: motivation, expertise, composition, team mental models, communication, cooperation, coordination of execution, and shared awareness. Patterns may be selected to counter or enhance particular forces, while the consequences may affect other components of the IPO model.

2 Task Management for Supervisory Control

We will be discussing the C2 of teams of Unmanned {air, surface, subsurface, ground} Vehicles (UxV). It is important to distinguish between different instances of human-autonomy teaming that will be involved in describing the Intelligent Multi-UxV Planner with Adaptive Collaborative Control Technologies (IMPACT) project. Per our approach for pattern descriptions [14] we will do this is by focusing on the different Work Objectives (WObjs) and Work Processes (WProcs).

Through the IMPACT project we are experimenting with concepts necessary to develop a C^2 capability to allow a small number of people to control multiple teams of autonomous vehicles. The WObj of the overall system is therefore to achieve success for the various missions that are taken on by the commander responsible for the UxV teams. There is a supervisory relationship that is formed between the controllers and UxV that are conducting the missions. This relationship is mediated through IMPACT.

In this paper, we will focus on the subordinate human-autonomy relationships that we believe enable the human controllers to handle the workload of a complex environment that could include many teams of many vehicles. We are not focused on the tasks and conduct of the vehicles or vehicle teams themselves, but rather the tasks of the controllers in maintaining C^2 over the vehicle teams. The use of autonomous assistants has been suggested for this class of problem before [15].

2.1 Task Generation

The IMPACT Task Manager (TM) is currently being developed, but exists as a functioning prototype with partial functionality. Some of the capabilities discussed will be those that are still in development, but the patterns exist in previous work as well, for instance in the Cognitive Assistant that Learns and Organizes [15] and other experiments in the use of task learning [16]. This is an important feature of patterns. They show up frequently.

TM monitors its environment, learns to recognize the need for and generate tasks, manages active tasks and assists the user in completing tasks. The TM is instrumented in order to improve performance. The WObj for the human-autonomy team is to execute the necessary C^2 tasks that will allow the command to succeed at its mission.

TM is programmed to recognize the need for some tasks. For instance, TM monitors a chat stream and recognizes text patterns that indicate the need to generate a new instance of a particular task type. Other tasks can be generated through machine learning that occurs through observing the human user and relating the user's actions to the current state of the C^2 environment. The human's actions serve to label the situation to enable lifelong learning by the cognitive assistant. Other user actions, such as correcting a mistakenly generated task, serve to teach the TM.

2.2 Task Assignment

Tasks that are created by the TM on its own, or through the human's initiative are then assigned for execution. The TM can choose either an autonomous assistant or a human, or a team of humans and autonomy to execute the task. The key to this flexibility is the task structure [5] that breaks tasks into methods and subtasks. Instrumentation can collect data on performance allowing TM to predict how each agent (human or machine) will perform.

Working agreements established by the users dictate how much authority the TM has in assigning tasks. The human users also have the ability to manually change the assignments that are created by the TM.

2.3 Task Execution

For some tasks, the TM will have aids that can help a user through the task. These can be programmed for tasks that are anticipated. For those that are not, we envision using capabilities similar to those in [15] for automated assistants to learn to help or perform tasks. When capable of performing tasks the methods are entered into the task structure and the assistant will be available to be assigned a task or subtask.

3 Patterns

We will discuss three patterns that we would propose are being used to allow human-autonomy teaming within IMPACT between a human user and the TM. Two of these patterns illustrate a heterarchical rather than a hierarchical relationship, suggesting that we do not have a supervisory control situation. Instead, humans and autonomous agents are teaming roughly as partners, though the roles do give added authority to the human. The final pattern discussed also creates a temporarily strict supervisory control authority. Using a template inspired by software design patterns (e.g., [8]), and the top-level patterns discussed in [14], we describe these patterns.

3.1 Instructor and Student

Figure 1 indicates the top-level pattern from [14] that is relevant for this aspect of TM. Both the TM and the human user are identifying tasks from the environment. The TM is accomplishing this by reading data from the C2 system and performing either programmed or learned pattern recognition. The human user is presumably doing pattern recognition as well, but we will not address this aspect of human cognition here. Either is able to initiate tasks in the system, which are subsequently assigned to an agent (either human or machine).

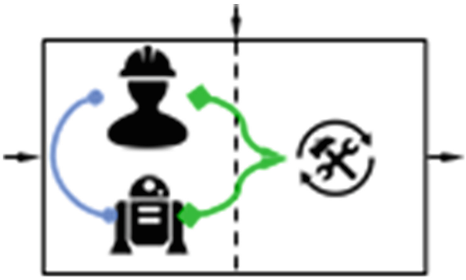


Fig. 1. Agent works in cooperation with the human operator being an element of the worker per [14].

Intent: Provide a means for expanding the pattern recognition and task catalog of the TM.

Motivation: We do not know a priori all of the tasks that will need to be accomplished. C^2 has been largely a human endeavor in part because it is difficult to discretize the tasks and events in the C^2 environment. Our programs do not know as well as human experts do, when tasks should be initiated. We can perform task analyses to expand our knowledge and influence our design, but the nature of the domain requires the ability to adapt.

Applicability: Instructor-Student is applicable whenever there is a means to represent a discretization of the environment and an opportunity for the human to indicate the nature of an element of the environment. There must be instrumentation in place to facilitate representation of the environment. The representation then allows us to form a useful model of the patterns required for identifying the discrete element (e.g., a task).

Structure and Participants: See Fig. 1 for the basic structure. The human is in the role of *instructor* while the cognitive agent is in the role of *student*. Both use the tools available in the system to gain situational awareness and to instantiate discretized elements (tasks).

Human Requirements and Consequences: This pattern requires effort and motivation by the human participant. If because of stresses the human does not take the opportunity to teach the agent, then the agent will not be able to assist through recognizing instances for task creation. One likely cause of this decision will be an inadequate learning environment for the agent. If either the instrumentation is insufficient, or the algorithms used produce poor models, then the user is likely to lose trust (rightly) in the ability of the agent to learn. This pattern when successfully implemented helps form a consistent team mental model. The student (autonomy) learns to classify events by creating a model that estimates the model held by the human instructor. Output performance measures of quality and quantity of tasks completed form the ultimate measure of the instructor-student team. The performance consequences are discussed in [16] and depend on the human's teaching strategy and the machine learning capabilities of the agent.

3.2 Working Agreements

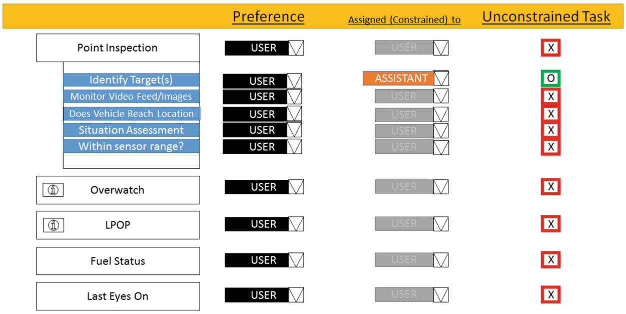
This aspect of TM also follows the top level pattern depicted in Fig. 1. The TM is serving as a scheduler for human and automated agents.

Intent: Provide a means for a human user to indicate preferences concerning how task allocation and execution should be accomplished.

Motivation: We want to establish the rules by which the allocation of tasks will be conducted. This allows the human user to successfully predict the pattern of operations increasing trust in the autonomy. It also puts the human on at least equal footing with the autonomy. The autonomy has algorithms, models, and sensors to use to make a decision as to how to assign tasks. This provides a means for the human to contribute to the decisions.

Applicability: Working agreements are applicable whenever two agents must cooperate and have differing kinds of models for making decisions. The working agreement bridges the mismatch in knowledge and understanding with agreed upon guidelines [19].

Structure and Participants: See Fig. 1 for the basic structure and Fig. 2 for a possible user interface. Both the human and the autonomy access the working agreement in



	Preference	Assigned (Constrained) to	Unconstrained Task
Point Inspection	USER	USER	X
Identify Target(s)	USER	ASSISTANT	O
Monitor Video Feed/Images	USER	USER	X
Does Vehicle Reach Location	USER	USER	X
Situation Assessment	USER	USER	X
Within sensor range?	USER	USER	X
Overwatch	USER	USER	X
LPOP	USER	USER	X
Fuel Status	USER	USER	X
Last Eyes On	USER	USER	X

Fig. 2. A possible user interface for accessing a working agreement

order to guide decisions. In all efforts that we are aware of, the agreement is between a human and an autonomous agent, although it is clearly modeled after human-human agreements.

Human Requirements and Consequences: This pattern brings about a team mental model concerning how work will be accomplished [17]. It also affects how team composition is adjusted for individual tasks. An agent's (particular human's) motivation is a source of information for the input of an agent to the working agreement. Working agreements can directly affect the quality and quantity of work performed and therefore measuring these factors can be useful. We also believe that working agreements increase trust in the automation by the humans involved as it gives the autonomy more predictable behavior based on a model accessible to the humans.

3.3 Ultimate Override

This aspect of TM takes a heterarchical relationship and imposes a hierarchy onto it. Figure 3 shows the base pattern graphically.

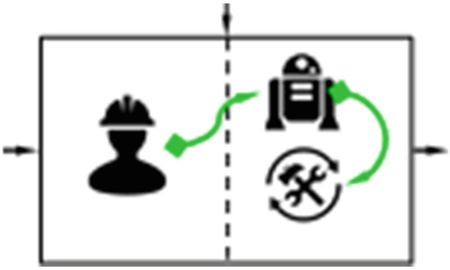


Fig. 3. The human user assumes all of the initiative for the WProc

Intent: Provide a means for a human user to take full control over a WProc. When the operator performs certain operations, there is no negotiation with the agent and any working agreement is overlooked.

Motivation: There are times when we want to ensure that the human has the final say. In many situations the human have the legal responsibility over WProc and WObj, while the automated system does not. Knowing that this option exists may give the human teammate more trust in employing the automation. However, it is also possible that in providing such manual overrides, users may under trust the autonomy [20].

Applicability: This pattern is applicable wherever there are legal or ethical guidelines that insist upon human responsibility. It is also applicable any time when the models available for the autonomy are known to be insufficient and human decision-making may become essential.

Structure and Participants: See Fig. 3 for the basic structure. Capabilities that employ this pattern are fundamentally taking the team away from Fig. 1 to a structure where the human exerts all of the initiative and the autonomy performs tasks assigned using the tools available.

Human Requirements and Consequences: This pattern removes all semblance of a team mental model. The initiative is completely contained within the human participant. This also changes the team composition and view of participant expertise. The autonomy essentially becomes a narrow technical expert, while the human moves to a supervisory role. The motivation of the human participant is all important in this pattern, while the autonomy essentially does not exhibit motivation (Fig. 4).

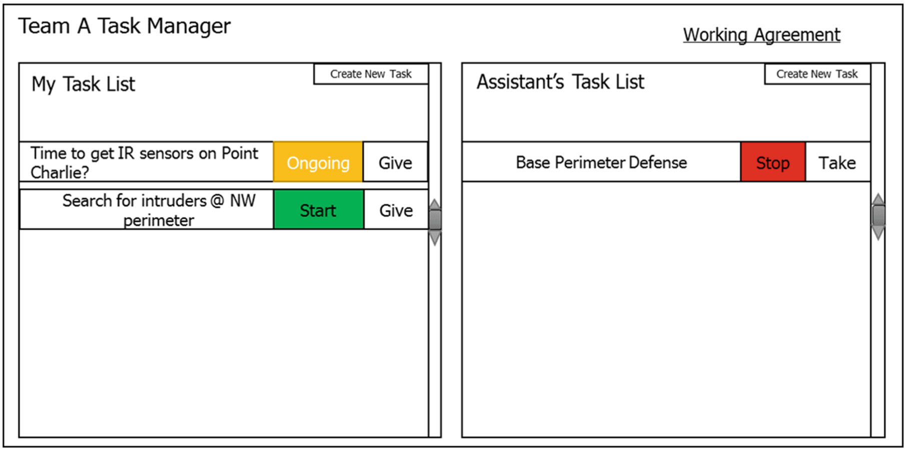


Fig. 4. When the user decides to "Give" or "Take" tasks while using the TM, ultimate override is being exerted. For that period and relating to those tasks, the TM makes no decisions, but reflects the motivated will of the supervisor.

4 Conclusions

We have described three patterns that we are employing in the development of a TM to support the C2 of teams of autonomous vehicles. We utilize the systems engineering approach from [14] along with the descriptive templates of sources such as those used in the domain of software engineering [8]. Alexander [6] would have us utilize an extensive catalog of such patterns and weave a specifying description of our projects developing a pattern language.

Further work must be done to fully develop such a catalog, and we plan on employing experiments to measure the human and team performance consequences of our use of these patterns. We must also be conscious of the possibility that consequences will differ across domains, and believe that research can enlighten us in this area. Ultimately, the core patterns of human-autonomy teaming will be those that can be reliably used across many application domains.

References

1. Willard, R.F.: Rediscovering the art of command & control. In: Proceedings of the US Naval Institute, vol. 128, no. 10 (2002)
2. Lange, D.S., Lai, E.C., Carlin, M., Ling, A.S., Keifer, K., Deans, B., Nitz, K., Tam, L., Bolton, J., Graves, B., Reestman, B.: SKIPAL: the incorporation of machine learning technology into the strategic knowledge integration web. In: The 15th International Command and Control Research and Technology Symposium (2010)
3. Apker, T.B., Johnson, B., Humphrey, L.: LTL templates for play-calling supervisory control. In: American Institute of Aeronautics and Astronautics SciTech Forum (2016)
4. Behymer, K.J., Mersch, E.M., Ruff, H.A., Calhoun, G.L., Spriggs, S.E.: Unmanned vehicle plan comparison visualizations for effective human-autonomy teaming. In: The 6th International Conference on Applied Human Factors and Ergonomics (2015)
5. Lange, D.S., Gutzwiller, R.S., Verbancsics, P., Sin, T.: Task models for human-computer collaboration in supervisory control of teams of autonomous systems. In: The IEEE International Inter-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (2014)
6. Alexander, C.: The Timeless Way of Building. Oxford U.P., New York (1979)
7. Alexander, C., Ishikawa, S., Silverstein, M.: A Pattern Language: Towns, Buildings, Construction. Oxford U. P., New York (1977)
8. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading (1977)
9. Fowler, M.: Analysis Patterns: Reusable Object Models. Addison-Wesley Professional, Reading (1996)
10. Pree, W.: Design Patterns for Object-Oriented Software Development. Addison-Wesley Professional, Reading (1994)
11. Brown, W., Malveau, R., McCormick, H., Mowbray, T.: Anti Patterns: Refactoring Software, Architectures, and Projects in Crisis. Wiley, Chichester (1995)
12. Coplien, J.O.: Pattern Languages of Program Design. Addison-Wesley Professional, Boston (1995)
13. Juziuk, J.: Design Patterns for Multi-Agent Systems. Linnaeus University, Sweden (2012)

14. Schulte, A., Donath, D., Lange, D.S.: Design patterns for human-cognitive agent teaming. In: Harris, D. (ed.) EPCE 2016. LNAI, vol. 9736, pp. 231–243. Springer, Heidelberg (2016)
15. Myers, K., Berry, P., Blythe, J., Conley, K., Gervasio, M., McGuinness, D., Morley, D., Pfeffer, A., Pollack, M., Tambe, M.: An intelligent personal assistant for task and time management. AI Mag. **28**(2), 47–61 (2007)
16. Lange, D., Carlin, M., Ivanchenko, V., Luqi, B.V.: Human and software factors for successful system adaptation. In: The Twelfth International Command and Control Research and Technology Symposium (2007)
17. Gutzwiller, R.S., Lange, D.S.: Task teams: supervisory control and task management of autonomous unmanned systems. In: Lackey, S., Shumaker, R. (eds.) VAMR 2016. LNCS, vol. 9740, pp. 1–9. Springer, Heidelberg (2016)
18. Gutzwiller, R.S., Lange, D.S., Reeder, J., Morris, R.L., Rodas, O.: Human-computer collaboration in adaptive supervisory control and function allocation of autonomous system teams. In: Shumaker, R., Lackey, S. (eds.) VAMR 2015. LNCS, vol. 9179, pp. 447–456. Springer, Heidelberg (2015)
19. de Greef, T., Arciszewski, H., Neerincx, M.: Adaptive automation based on an object-oriented task model: implementation and evaluation in a realistic C2 environment. J. Cogn. Eng. Decis. Mak. **4**(2), 152–182 (2010)
20. Parasuraman, R., Riley, V.: Humans and automation: use, misuse, disuse, abuse. Hum. Factors **39**(2), 230–253 (1997)