The Playbook™ Approach to Adaptive Automation

Christopher Miller, Harry Funk, Peggy Wu, Robert Goldman, John Meisner, Marc Chapman
{cmiller, hfunk, pwu, rpgoldman, jmeisner, mchapman}@sift.info
Smart Information Flow Technologies, Minneapolis, MN

Abstract

SIFT has pioneered a human-automation integration architecture, called Playbook™, based on a shared model of the tasks in the domain. This shared task model provides a means of human-automation communication about plans, goals, methods and resource usage—a process akin to referencing plays in a sports team's playbook. The Playbook enables human operators to interact with subordinate systems with the same flexibility as with well-trained human subordinates, thus allowing for adaptive automation. We describe this approach and its application in an ongoing project called Playbook-enhanced Variable Autonomy Control System™ (P-VACS).

Introduction

As automation becomes more sophisticated, managing the human-machine interface becomes more complex. Easy to use human-automation interfaces analogous to human-human delegation are limited to simple tasks. For complex operations, there is often the danger that the automation is simply transferring workload from one task to another (e.g. supervision of automation), or even adding to the user's workload or attentional demand (Bainbridge, 1983). Elements of trust and etiquette (see Miller, 2004) undoubtedly play a large role in the way we use automation, as does the choice of tasks to automate (Parasuraman et al., 2000). There is a substantial body of guidelines for creating effective human-automation interactions, but most are abstract and there is no consensus on how to implement guidelines into design (Mitchell, 1998). One concept for reducing human workload is the creation of adaptive systems as opposed to adaptable systems (from Oppermann, 1994). The chief distinction between the two is that an adaptable system allows the user to make his/her own changes, whereas an adaptive system must make its own decisions about the adaptations to be made (Funk and Miller, 2001). However, regardless of whether a system is adaptable or adaptive, one specific challenge in the implementation of any human-automation interaction system lies in creating an underlying representation for the user and the automation to communicate about tasks, resources, and intent. We describe our architecture and the technologies for an adaptive system as applied in an ongoing project called Playbook™-enhanced Variable Autonomy Control System™ (PVACS), a multiple Unmanned Aerial Vehicle (UAV) control system.

Task Representation

Supervisory control is a design concept for enhancing the effectiveness of human-automation interaction. In supervisory control, operators select tasks for automation and provide instructions for how to perform them. Tasking of a high level plan is equivalent to expressing intent about who is to perform which of the sub-tasks in that hierarchy and in what way. Real time supervisory relationships with automation have rarely approached the flexibility of effective human-human delegation, but substantial research shows that enabling such relationships would provide important benefits, including improved situation awareness, more accurate usage decisions, balanced mental workload, increased user acceptance, and improved overall human + machine performance. The challenge in providing such a task delegation mechanism is to make it possible for the operator to express his or her intent to the automation in a way that (a) is quick and easy enough to be feasible in an operational setting, (b) is comprehensible by all parties, and (c) will consistently and reliably represent the intent and constraints given highly diverse situations.

Some members of our team have pioneered this view (Miller and Goldman,1997; Miller et al., 2000, Goldman et al., 2000; Miller and Parasuraman, 2004), and others are coming to the same conclusion (Bonasso, 1999; Shrekenghost, 1999; Cruz et al., 2001; Tabuada et al., 2001). We have created proof of concept illustrations of "Tasking Interfaces" which enable a human to express his or her intent to an automated controller at various levels of a hierarchical decomposition of tasks. This is important as *variable autonomy control* allows the human operator to stipulate or prohibit (constrain) the use of specific methods or resources at all of the various levels. The human user of such a system can express high-level mission goals or very specific mission plans—or anything in between. These have been shown to be the primary components of good intent specification in human-human communication in military command and control domains (Klein, 1998; Shattuck, 1995). This flexibility to express intent at various levels of abstraction is perhaps better aligned with how we interact with each

---

™ Playbook is a trademark of Smart Information Flow Technologies

™ Variable Autonomy Control System and VACS are trademarks of Geneva Aerospace

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **2005**	2. REPORT TYPE	3. DATES COVERED **00-00-2005 to 00-00-2005**

4. TITLE AND SUBTITLE **The Playbook (TM) Approach to Adaptive Automation**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Smart Informaiton Flow Technologies (SIFT) LLC,311 First Street North Suite 300,Minneapolis,MN,55401**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**5**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

other (Cockburn). Furthermore, structured text and speech-act related patterns as a means of restricting user interfaces have been shown to be beneficial to collaborative tasks (Matesssa, 2001).

One key to creating automation that is smart enough to make instructing easy, yet subservient enough so that it always adheres to the operator's intent, is the establishment of a common understanding between the human operator and the automation. SIFT has pioneered a human-automation integration architecture, called Playbook, based on a shared model of the tasks in the domain for the purpose of achieving common understanding between components (both human and machine). This shared task model addresses the challenge mentioned above by providing a means of human-automation communication about plans, goals, methods and resource usage—a process akin to referencing plays in a sports team's playbook. Playbook is a specific method of implementing a delegation interaction and can be roughly divided into two components, (1)a hierarchical task model that is compatible with levels of automation (cf. Sheridan, 1987), and (2) a planning mechanism for evaluating existing resources, plan validity, and instantiating the task models. Below we compare current UAV control methods to the Playbook method.

In the current state-of-the-art UAV control systems, each UAV requires at least one fully trained pilot with knowledge of most or all UAV sub-systems. When initializing a mission, the operator must manually plot the flight path, a process that may take 30 minutes before the UAV is launched. The UAV then needs to be controlled via joystick commands at a remote control station for the duration of its flight. The automation is limited to the translation of a small set of user commands to controls for actuators. The automation does not have the capability to model the operator's intent, nor store complex commands. This presents an opportunity to introduce a shared task model. By identifying a set of common tasks, grouping them into plays, and parameterizing elements such as time and location, a set of *play templates* can be created. When a previously defined play is to be executed, the operator can select a play template and bind the parameter values as appropriate to his/her needs. Both the operator and the automation have a similar model of the sequence of tasks to execute.

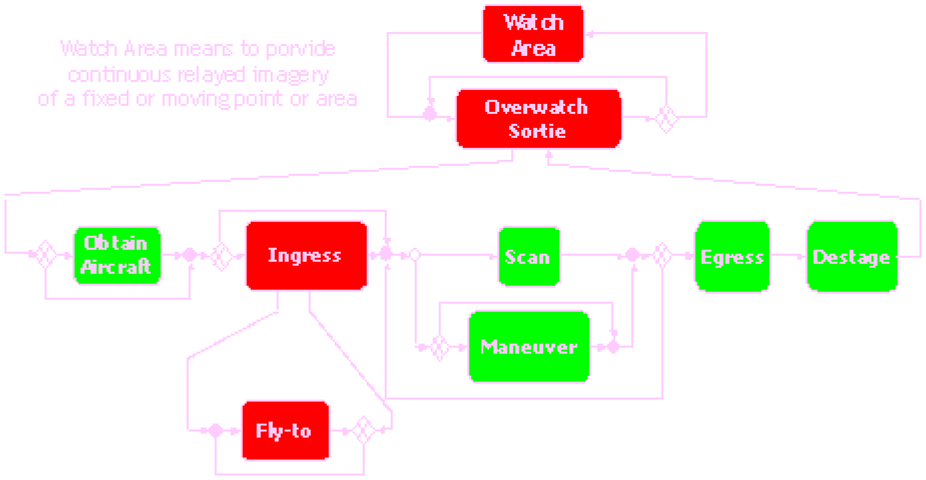

Figure 1 A template for the Watch Area play.

Figure 1 illustrates an example of such a play template. In this example, the Watch Area play contains at least one sub-play called "Overwatch Sortie". The significance of the diamond following Overwatch Sortie is that after performing an Overwatch Sortie, the parent play ("Watch Area") may iterate back through one or more subsequent instances of Overwatch Sortie, or it may not—a single instance may be sufficient to complete the play. Drilling down further in the representation of Watch Area, we see that each instance of Overwatch Sortie may require an initial step of "Obtain Aircraft" (a task that can be satisfied by various methods including requesting an idle resource or removing one from an active, yet lower priority play). Next, an Ingress task may be necessary and, if so, it will be composed of one or more "Fly-to" waypoint legs. After Ingress is complete for the vehicle associated with this Sortie, the vehicle will Scan and may also Maneuver. These actions may repeat until some condition (generally, the time requirements for the parent Watch Area task) are completed. Following Maneuvering and Scanning, the Sortie includes an Egress and a Destage task.

Since the set of procedures is now represented in the automation, the operator may call this play and provide the necessary parameters each time s/he wishes to execute a mission that resembles "Watch Area". UAV control is particularly suited to use such a strategy because some form of a common task model already exists in order to facilitate communication between human operators for UAV controls that require multiple operators. By introducing this abstraction layer, a set of platform independent play templates can be created so that the operator can call the same play regardless of the specific type of UAV. Furthermore, s/he now has the ability to dynamically ‘tweak’ the mission, such as designate threat areas or adjust plans without regard for the UAV operation.

Adaptive Automation and Relaxed Planning

As described above, the ‘playbook’ contains predefined patterns of behavior that are understood by all participants on the team (including the automation). By means of a single, short play name or label, the operator can express his/her intent for a large number of independent actors to behave in a dynamically-changing yet coordinated and effective fashion. Plays can serve as a shared point of reference from which to build novel variations with minimal effort. It is worth noting that plays also require the actors to be capable of interpreting and applying the play to the context in which the play is called. In a sports playbook analogy, this may be as simple as deciding whether to step left or right depending on what direction the opponent is coming from in a football play, but it precludes complete rote behavior. Actors must be allowed some autonomy about how to perform their delegated roles if there is to be any efficiency gain in the system.

Therefore, creating a large database of play templates is only part of the solution. The flexibility of these play templates are highly limited by the automation’s ability to model the user’s goals and intents. Suppose an operator wants to perform a portion of Time Sensitive Targeting task, say, performing surveillance on a target, beginning at a specific time, and for a particular duration. The automation may check the availability of resources and find that no UAVs are available at the specified start time. If the automation has no additional models of the operator’s intent, it may stop searching, and report that such a plan cannot be created. A human subordinate, however, may understand that surveillance for some period is better than no surveillance at all, realize that adequate resources will become available 5 minutes later than the specified start time, and offer a ‘relaxed’ plan as an alternative. By building this knowledge into the automation, it is able to interpret and apply a play in context, allowing for more flexibility and efficiency, resulting in an adaptable system closer to a skilled human subordinate.

We addressed this issue by structuring our architecture so that such knowledge can be incorporated into the automation, but is abstracted from the task models. The overall Playbook architecture consists of three components, as presented in Figure 2:

1. A library of task models
2. A constraint-based planning engine
3. A User Interface

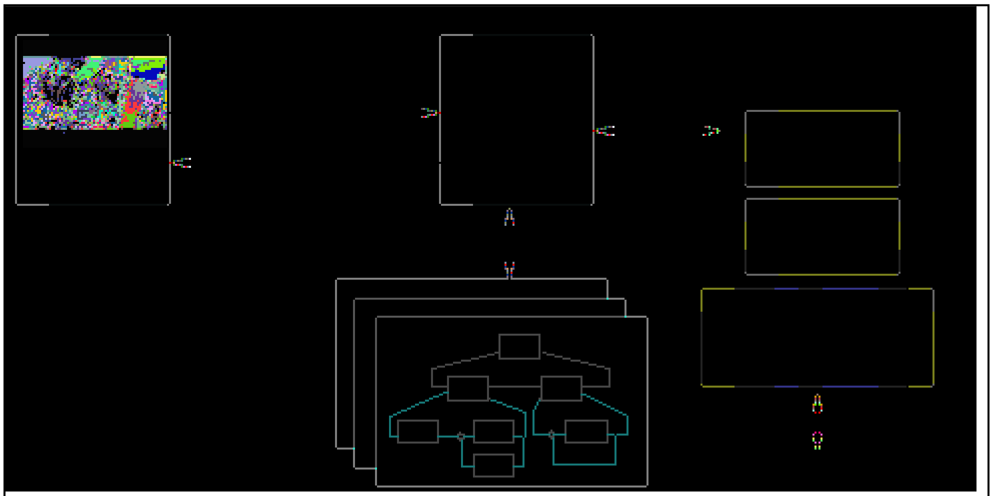

Figure 2 General Playbook Architecture.

As described above, the hierarchical structure of task models is used to represent intent, which is decomposed into subtasks. When the user selects a play and specifies constraints, the planning component, through its own knowledge of viable task structures and through interaction with sophisticated simulation tools (in this case, the VACS execution environment), is capable of both fleshing out a plan within specified parameters and of critiquing a plan for feasibility and goal accomplishment. The planner accomplishes this by access to knowledge about the resources (for example, quantities such as fuel or munitions, as well as less obvious resources such as time, distance, or human attention and cognition capabilities) used by specific tasks in the scenario, and knowledge of how *legal* task combinations are known to accomplish goals.

Whenever known resource violations occur, the planner can report that this is not a feasible plan. Similarly, whenever task combinations do not *add up* to the accomplishment of a parent goal, the planner reports the conflict. The planner may operate in one of two modes. The first is the critiquing mode, where it will simply report the conflict. The second, more complex mode is the autonomous planning mode, where the planner will choose another method for accomplishment and present its improvisations to the operator. The operator is then presented with a choice to accept or reject the planner's suggested plan. This planning capability is not a full simulation, but rather a first-pass, coarse-grained *constraint checking* capability. It does not, for example, contain any ability to simulate world states or enemy actions. Nevertheless, it is a useful method of doing some plan generation and screening for obvious errors. When, as in the Time Sensitive Targeting task described in the example above, the tasks in the plan are to be performed by humans, this level of planning is perfectly adequate – it allows the humans involved some flexibility in the specifics of performing the tasks, while maintaining synchronization and communication when multiple human actors are involved.

Playbook by itself is an environment for human interaction and planning and does not include the event handling and control algorithms necessary to execute missions on real vehicles. As in Figure 2, Playbook must be integrated with a control architecture that provides these capabilities. Geneva Aerospace's Variable Autonomy Control System (VACS) provides a robust integrated control architecture enabling a single operator to control multiple UAVs (Duggan, 2001). The VACS architecture links teams of UAVs with remote operator workstations, where a human operator must make all mission level decisions and interact with the various control levels.

The Playbook integration advances VACS to higher levels of autonomy by providing automated means of developing and adjusting plans to achieve mission objectives. Playbook possesses a hierarchical understanding of the operational intent and specific target tasking, and can provide high-level commands to the vehicle and sensor control systems following the command structure already in place in the VACS. In essence, VACS provides a "library" of control execution behaviors from which increasinly complex sequences of tasks can be composed into plays. The integrated Playbook + VACS (PVACS) capabilities are particularly relevant to operations where busy and/or non-rated operators must supervise multiple or heterogeneous vehicles. PVACS' combination of very high level and variable autonomy control allows busy operators to command sophisticated, coordinated behaviors simply and rapidly and allows operators with more time or training to impose highly specific commands to customize vehicle behavior to their exact needs.

Additional details about the playbook concept can be found in (Miller and Goldman, 1997) and (Miller and Parasuraman, 2004). More detailed information about one version of the Playbook's reasoning and planning component can be found in (Goldman et al., 2000), though we are currently at work on improving that reasoning component and its knowledge representation. A more detailed presentation of the PVACS prototype and description of the user interface and user interactions with it can be found in (Miller et al., 2000).

Value of Playbook

In previous research we have obtained empirical evidence for the efficacy of Playbook type interfaces for mission efficiency when a single operator has to supervise multiple agents (Miller and Parasuraman, 2003; Parasuraman et al., 2003; Squire et al., 2004). We used the RoboFlag simulation platform (see Parasuraman et al., 2003) with a simplified Playbook interface to emulate a typical Unmanned Vehicle (UV) mission involving a single operator managing a team of up to 8 agents. The results showed that the multi-level tasking provided by the Playbook interface allowed for effective user supervision of agents, as evidenced by the number of missions successfully completed and the time for mission execution. In addition, the flexible Playbook interface was superior to fixed control conditions in which the operator had access only to either direct control of individual agents or automated plays alone, but not both. Finally, the superiority of the flexible Playbook interface was particularly apparent in conditions when the opponent posture was unpredictable. These findings provide strong support for the view that the Playbook allows for effective tasking of multiple agents while keeping the supervisor in the decision-making loop, without increasing supervisor mental workload, and allowing the human supervisor to adapt successfully to unpredictable changes in the environment. These benefits are important because traditional human-agent interfaces have often been found to result in significant system and human performance costs—including mode errors, user under- and over-reliance on automation, and reduced situation awareness (Parasuraman and Riley, 1997;Parasuraman et al., 2000). Such limitations are sometimes severe enough to result in catastrophic accidents, as evidenced by numerous analyses of aviation incidents, including unmanned aircraft such as the Air Force's Predator (Parasuraman and Byrne, 2003). Hence, the development of appropriate human-automation interfaces is critical for effective human supervision of autonomous agents. Playbook provides such an interface concept. Its benefits will be particularly apparent in situations of environmental uncertainty and where unexpected events occur, making pre-programmed automated behaviors ineffective.

Future Work

Far more than 'just' user interfaces, Playbook provides a complete architecture for the integration of human input, intelligent *a priori* planning, reactive planning and event handling, and ongoing vehicle control loops. To date, development on this tasking interface architecture has been directed at ground-based control of remote vehicles, and at a priori mission planning. However, our general tasking interface architecture extends to work with software components and is not limited to the vehicle control domain. SIFT is pursuing the application and extension of Playbook in a number of different directions. One particular direction is in developing methodologies to build more extensive task models, such as the ability to derive Playbook task knowledge from results of Cognitive Work Analysis (CWA) of a task domain and then use the Playbook architecture (including UI and planning components) to produce useful task timeline inputs for a constructive simulation. Thus far, our emphasis in developing a representation has not been on computational efficiency or even on specific software representations, but rather on ease of accurately and comprehensively expressing knowledge requirements.

Acknowledgements

The PVACS program is funded by a Phase II Small Business Innovation Research grant from DARPA IXO, administered by the U.S. Army Aviation & Missile Command, contract number DAAH01-03-C-R177. We would like to thank Dr. John Bay for his oversight and advice during the investigation. We are also grateful for the work of Geneva Aerospace personnel, especially Billy Pate, on the VACS system and for educating us about it.

References

Bainbridge, Lisanne (1983). Ironies of Automation. Increasing levels of automation can increase, rather than decrease, the problems of supporting the human operator. Automatica, 19, 775-779. Reprinted in : (1987) Rasmussen, J., Duncan, K. and Leplat, J. (eds.) New Technology and Human Error, Wiley, Chichester, pp. 276-283. http://www.bainbrdg.demon.co.uk/Papers/Ironies.html

Bonasso, P. Issues in providing adjustable autonomy in the 3T architecture. Working Notes of the AAAI Spring Symposium on Agents with Adjustable Autonomy. Stanford, CA; March 22-24, 1999.

Cockburn, A. "Growth of Human Factors in Application Development" http://alistair.cockburn.us/crystal/articles/gohfiad/growthofhumanfactorsinsd.htm

Cruz, J., Simaan, M., Gacic, A., Jiang, H., Letellier , B., Li, M. and Liu, Y. "Game-Theoretic Modeling and Control of a Military Air Operation," IEEE Transactions on Aerospace and Electronic Systems, Vol. 37, No.4, October 2001.

Duggan, D. S. (2001). "Demonstration of an Integrated Variable Autonomy UAV Flight Control System", Phase II SBIR Final Report, AFRL-HE-WP-TR-2001-0035, January.

Funk, Harry B. and Christopher A. Miller (2001). User Acceptance and Plan Recognition: Why Even Perfect Intent Inferencing Might Not be Good Enough. In Prceedings of 2001 AAAI Fall Symposium.

Goldman, R., Haigh, K., Musliner, D., Pelican, M. (2000). MACBeth: A Multi-Agent Constraint-Based Planner, In Working Notes of the AAAI workshop "Constraints and AI Planning", 2000. 11-17.

Klein, G. Sources of Power. MIT Press; Cambridge, MA. 1998.

Matessa, Michael (2001). Structured Interfaces in Collaborative Communication. In Prceedings of 2001 AAAI Fall Symposium.

Mitchell, Christine (1998). http://mentalmodels.mitre.org/cog_eng/reference_documents/OFM-Model%20Based%20Design.pdf

Miller, C. (Guest Ed.) (2004). "Human-Computer Etiquette: Managing Expectations with Intelligent Agents." Communications of the Association for Computing Machinery. April.

Miller, C. and Goldman, R.(1997). 'Tasking' Interfaces: Associates that know who's the boss. Proceedings of the 4th USAF/RAF/GAF Conference on Human/Electronic Crewmembers, (Kreuth, Germany, September 1997).

Miller, C. and Parasuraman, R. (2003). Beyond Levels of Automation: An Architecture for More Flexible Human-Automation Collaboration. Proceedings of the 47th Annual Meeting of the Human Factors and Ergonomics Society, October 13-17; Denver, CO.

Miller, C. and Parasuraman, R. (2004)"Tasking" Interfaces for Flexible Interaction with Automation: Playbooks for Supervisory Control. IEEE Transactions on Systems, Man and Cybernetics; Part A: Systems and Humans.

Miller, C., Pelican, M. and Goldman, R. (2000). "Tasking" Interfaces for Flexible Interaction with Automation: Keeping the Operator in Control. Proceedings of the Conference on Human Interaction with Complex Systems. (Urbana-Champaign, Ill. May 2000).

Oppermann, R. (1994). Adaptive User Support. Lawrence Erlbaum: Hillsdale, NJ.

Parasuraman, R., & Byrne, E. A. (2003). Automation and Human Performance in Aviation. Principles of Aviation Psychology. P. Tsang and M. Vidulich, Eds. Erlbaum, Mahwah, NJ, 2003, 311--356.

Parasuraman, Galster, & Miller, 2003, "Human Control of Multiple Robots in the Roboflag Simulation Environment", in Proceedings IEEE Systems, Man and Cybernetics (2003).

Parasuraman, Raja, Thomas B. Sheridan, and Christopher D. Wickens (2000). A Model for Types and Levels of Human Interaction with Automation. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, Vol. 30, No. 3, May 2000.

Parasuraman, R. and Riley, V., "Humans and automation: Use, misuse, disuse, abuse," Human Factors, Vol. 39, pp. 230-253, 1997.

Shattuck, L. Communication of Intent in Distributed Supervisory Control Systems. Unpublished dissertation. Ohio State University, Columbus, OH. 1995.

Shreckenghost, D. Human interaction with control software supporting adjustable autonomy. In Working Notes of the AAAI Spring Symposium on Agents with Adjustable Autonomy. Stanford, CA; March 22-24, 1999.

Peter Squire, Hiroshi Furukawa, Scott Galster, Christopher Miller, and Raja Parasuraman, " Adaptability and flexibility are key! Benefits of the " playbook " interface for human supervision of multiple unmanned vehicles, " the Proc. Human Factors and Ergonomics Society 48th Annual Meeting, New Orleans, USA and September (2004)

Tabuada, P., Pappas, G. and Lima P. Feasible formations of multi-agent systems. Submitted to the IEEE Transactions on Robotics and Automation, Mar. 01