Delegation to Automation: Performance and Implications in Non-optimal Situations

Christopher A. Miller[1], Tyler H. Shaw[2], Joshua D. Hamell[1],
Adam Emfield[2], David J. Musliner[1], Ewart de Visser[2], and Raja Parasurman[2]

[1] Smart Information Flow Technologies, 211 First St. N. #300
Minneapolis, MN USA 55401
{cmiller,jhamell,musliner}@sift.net
[2] Human Factors and Applied Cognition Program, George Mason University,
4400 University Dr MS3F5, Fairfax, VA USA 22030
{tshaw4,aemfield,edevisse,rparasur}@gmu.edu

Abstract. We have previously advocated *adaptable* interaction with automation through approaches derived from human-human delegation and using the metaphor of a sports team's "playbook". In work sponsored by the U.S. Army's Aeroflightdynamics Directorate (AFDD), we have been studying the effects of play-based delegation on human-machine system performance. Of particular interest is performance with plays in "non-optimal play environments" (NOPE) where no, or only poorly fitting, plays exist to achieve needed behaviors. Plays have been shown to offer benefits in situations for which they are customized, but more interesting is whether complacency, expectation, loss of training, and automation bias might affect performance when plays do not perfectly fit. We provide a taxonomy of NOPE conditions and report on the exploration of some of these conditions in a series of three experiments performed to date.

Keywords: adaptive/adaptable automation, playbook, delegation, automation complacency, automation bias, mixed initiative automation.

1 Introduction

We have previously advocated *adaptable* automation through flexible delegation and, the metaphor of a sports team's "playbook" (e.g. [1]). In adaptable automation, the human initiates behaviors by "delegating" desired goals, plans, constraints or stipulations at flexible levels of specificity, which automation is then responsible for executing. By contrast, more traditional *adaptive* automation approaches leave decisions about when and how to adapt to the automation. Playbook®, SIFT's approach to adaptable automation for uninhabited vehicles (UVs) allows humans to "call a play" which an automated planning and execution control system understands. The planning system then expands that play to an executable level and manages its execution, replanning as necessary within the constraints imposed in the initial play calling.

Plays are not scripts, but are templates of goals and partial plans that must be instantiated for existing circumstances when called. Plays can be, and in our work have been, represented by hierarchical task networks which embody alternate methods of

D. Harris (Ed.): Engin. Psychol. and Cog. Ergonomics, HCII 2011, LNAI 6781, pp. 322–331, 2011.

performing the play. Plays can be called at a high level, in which case the operator delegates authority over all decisions which must be made about alternate subtask methods and resource usage decisions (within the "space" defined by the play definition itself) to the automation. Alternatively, the operator can "dive down" into the hierarchical structure of the play to offer increasingly specific "instructions" (constraints and stipulations) about exactly how a given instance of the play must be performed.

Previous work [2, 3, 9] has shown that flexible play-based delegation approaches provide payoffs in terms of human-machine performance across a variety of context conditions. In recent work sponsored by the U.S. Army's Aeroflightdynamics Directorate (AFDD), we have been using a multiple Unmanned Aerial Vehicle (UAV) simulation environment called MUSIM (for "Multi-UAV Simulation") to study the effects of play usage. Of particular interest has been "non-optimal play environment" (NOPE) conditions where the plays which exist provide no good fit for the circumstances. We might expect plays to offer benefits in situations for which they are customized since they offer a streamlined means of activating automation. More interesting, though, is whether complacency, expectation, loss of training, and automation bias might affect performance when plays do not fit. In this paper, we will define our use of plays, provide a taxonomy of NOPE conditions and report on the exploration of some of these conditions in a series of experiments performed to date.

2 Plays, Playbook® and Play Calling

A *play* (whether defined for automation or humans in teams) bounds a "space" of behaviors which are agreed to fall under the label of the play name. The behavioral space can be thought of as a hierarchical decomposition of alternate tasks, as in task analysis [4] and hierarchical task network planning [5]. The top level of this hierarchy represents a goal to perform the play, with various sub-goals that decompose the parent into alternate methods of accomplishment. Note, that the space does not include *all* possible behaviors the system can perform. Instead, only certain behaviors in certain combinations are agreed be exemplars of the play. For example, a "Hail Mary Pass" is a play (cf. Fig. 1) in American football in which many receivers run far downfield and the quarterback attempts to throw the ball to one. This play definition supports a wide variety of specific methods (e.g., exactly how many players run downfield, what patterns they run, when the ball is thrown, etc.) but some behavioral combinations fall outside the play definition. For example, a "Hail Mary Pass" in which zero or one player runs downfield to receive is non-sensical by definition.

There are three other important attributes of plays to be noted. First, plays can generally not be exhaustively defined in advance in a changing and incompletely knowable world. A quarterback will rarely specify a priori who he will throw the ball to, since that will be a function of who is least well defended. Second, plays demand that some autonomy be delegated to intelligent subordinates if any workload reduction and effective use of diverse skills is to be obtained. Third, the hierarchical play structure—especially its capturing of alternate methods to satisfy the play—provides a framework for conversation about exactly how the coach or quarterback intends an instance of the play to be performed—for example, stipulating how many and which receivers should go downfield and what patterns they should run to avoid confusion.

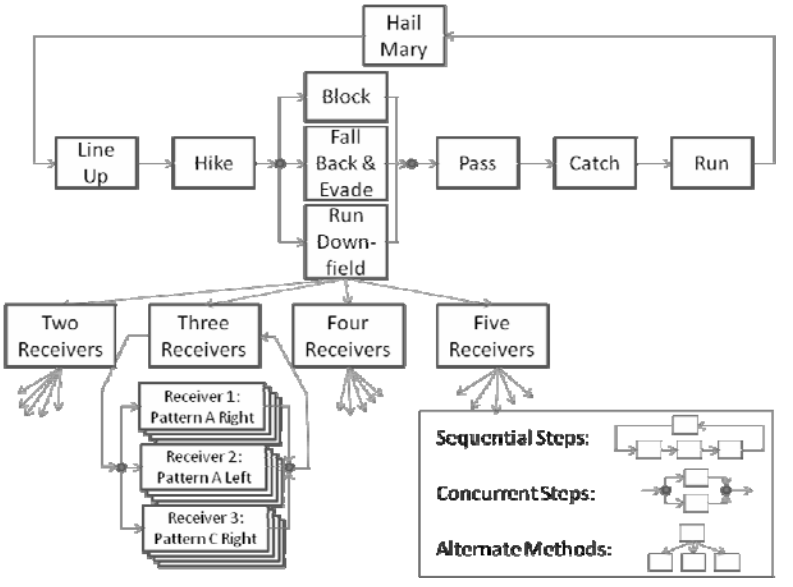

Fig. 1. Example task decomposition of a "Hail Mary" play-- showing sequential and concurrent dependencies as well as alternative methodologies

Play calling is a delegation method in which the supervisor's intent is expressed in a predefined play vocabulary. A "play" provides a label (the play's name) by which very complex behaviors can be activated quickly by a well-trained team, as well as a structure for further discussing and refining the instructed behavior. As described in prior work [1], play calling is a means of giving efficient flexibility to a supervisor over how much time s/he wants to spend in explicitly declaring intent vs. relying on subordinates' intelligence to achieve desired goals. But plays are always overlaid on a range of behaviors that is larger than that captured in the plays themselves. Players can do things, and do them in combinations, that are not be captured in the play set.

We have been developing human-automation interaction systems that provide delegation and play calling capabilities to human supervisors in interaction with smart subsystems—UAVs in most of our work. Our efforts have revolved around a core architecture we call *Playbook*® by analogy to the set of plays a sports team uses. Playbook is described elsewhere (e.g., [1]) but we will provide a brief description here.

In all Playbook architectures, the user communicates with automated systems through Playbook using a user interface (UI) configured to the domain and context of use. User instructions are interpreted by an Analysis and Planning Component which is a planning system. The UI is built around, and the Planning Component is designed to understand, a Shared Task Model which is a hierarchical decomposition of methods/plays defined. In current versions, the planner is built around a Hierarchical Task Network planner, SHOP2 [5]. The supervisor/user calls a play with whatever additional stipulations are desired via the UI. The planner then attempts to develop a plan (perhaps with existing, special purpose planners for routes, sensor coverage, etc.) which accomplishes the play in the current circumstances. If this is impossible, the

planner reports it and may begin a dialogue about what is feasible. Once a plan for executing the play is agreed upon, Playbook manages the execution, adapting it within the constraints of the play called. In current versions, execution monitoring is performed by a modified version of the OpenPRS procedure-based execution management system [6]. Playbook's executive sends ongoing commands to the real or simulated control algorithms of the UAV(s) it manages, and receives updates from them to iterate through this plan management process.

3 Non-Optimal Play Environments

Plays achieve their efficiency by compiling a set of behaviors from among all those possible and assigning an easily-accessed label to them. If every possible combination had a label assigned, determining the correct one would be inefficient for both supervisor and subordinates. Thus, the play set will need to be limited for most domains. In fact, a survey of current web forums discussing the number of plays in a football team's playbook turns up answers ranging from 10-12 for a junior team to around 100 for a professional team (depending on the manner of counting variations).

Plays will typically be defined for useful behavior which either recurs frequently or which, though rare, is anticipatable and will need efficient communication and coordinated. As such, plays will necessarily capture and label some combinations of behaviors at the expense of others. A well-designed play set will provide efficiency by making critical and/or repeatedly needed complex behaviors rapidly accessible, but it will nevertheless leave some less common, less anticipated contexts less well covered. That is, available plays will be "optimal" for some contexts in that they will be easiest to command, most readily understood and provide the most effective and accurate behavior from the subordinates. But the set will inevitably be "non-optimal" for others.

In this sense, defining plays is analogous to defining automation itself. Automation makes some tasks easier, but may make rarer, less expected tasks more difficult by taking the human "out of the loop" and making him or her subject to "automation bias", complacency, and skill loss [7,8]. Might play definition may be subject to similar perils? For example, might providing a play that proves generally useful in streamlining access to a pattern of automation behaviors make it more difficult for a supervisor to access individual behaviors and combine them in a novel fashion in rare circumstance for which the set of plays provides no useful coverage?

In this research, we were particularly interested in the contrast between human performance with Playbook automation in "Non-Optimal Play Environments" (NOPE). There are various ways in which play calling can be "non-optimal":

1. *No appropriate play exists*-- there is no way for the supervisor to declare what s/he wants in a language the subordinate understands. Even here, though, especially when the intermediate levels of hierarchically decomposed plays are accessible, other plays may be useful to perform part of the desired functionality.
2. *The play is hard to command*-- The user can declare what's desired, but doing so requires excess work because the declaration "language" is not efficient. For example, excessive options must be specified for activating the desired play version (e.g., excessive tuning or stipulation requirements). This will usually result when a

default (and most easily commandable) version of a play does not fit the current need, but can be modified or further constrained to be made to fit. Thus, it is a problem of play definition rather than of UI (see below).

3. *Play communication is poor*—The play set is a good fit for the contexts and goals, but the user has difficulty communicating them. In human-human interaction, this might be due to a failure of the supervisor to enunciate clearly, or a radio channel which is full of static. In human-automation interaction, the UI itself is the problem, not the play set or reasoning about it--e.g., excessive pull down menus rather than a direct graphical or speech commanding.
4. *The play is poorly executed*– That is, the supervisor can effectively communicate intent and the subordinate(s) can understand it, but they can't perform it reliably, either due to lack of knowledge or skill or both. In these circumstances, plays are not at fault, but attempting to use them may obscure the more fundamental flaws of the subordinate agents by implying that that functionality is commandable.

In the research reported below, we have primarily focused on the first of these NOPE types—conditions in which the set of plays available, though generally useful, is lacking a play for a set of conditions that arise. The other NOPE types remain important and of interest, but investigating them must await future research.

4 NOPE Experiments and Results

Previous experimental work has shown distinct benefits in overall performance and perceived human workload for delegation-based interaction systems [2,3,9], but this might be expected if plays were optimized for the conditions in the experiments. Here, we wished to examine conditions under which plays are not optimal for at least some of the conditions which occur—and in which the operator is required to abandon play usage and instead rely on more primitive behavior commanding.

We made use of the Multi-UAV Simulation (MUSIM) testbed developed by AFDD and illustrated in Fig. 2. MUSIM simulates control and imagery from multiple UAVs (notionally, Shadows) operating simultaneously. It provides both low level (joystick) flight and sensor control and somewhat more complex, autopilot-like capabilities such as waypoint control, simple flight patterns (such as circles and racetracks), and ground target tracking. We used a hybrid version of our Playbook and the Delegation Control ("DelCon") system developed by personnel at AFDD [3] to provide play-like delegation control of these lower level behaviors in MUSIM.

In the majority of our work, the MUSIM environment has been configured to provide three UAVs with slightly different capabilities operating in an urban environment to monitor three pre-designated locations ("Named Areas of Interest" or NAIs) for civilian, military and weaponized military vehicles. UAV Alpha can only provide camera imagery; Bravo can provide imagery and can lase targets to provide shooting coordinates—but cannot itself shoot; Charlie can provide imagery and can shoot, but cannot last (and therefore must coordinate with Bravo to shoot). The user's job is to manage these three UAVs to monitor vehicles moving about the urban area, accurately discriminate military threats from civilian vehicles (by "painting" civilians to designate them as such), and then track, lase and shoot them.

Plays in Playbook and DelCon were designed to assist with these tasks and are described below. In all cases, discrimination of civilian vs. military vs. weaponized ground vehicles had to be done visually by the participant, as did steering the sensor to center the camera on a vehicle and then putting that sensor in "track mode".

1. *Monitor NAI*—one user-specified UAV is routed to and set to orbit and provide video of one user-specified NAI (from the set of three pre-designated NAIs).
2. *Lase*—Given that a vehicle is already being tracked, task a (user-specified) laser equipped UAV to lase it and pass coordinates to a missile-equipped UAV.
3. *Monitor NAIs*—Each of the three UAVs is tasked (via this single play call) to move to orbit one of the three NAIs apiece and provide video of it.
4. *Track Target and Reconfigure*—Task a user-specified UAV to continue doing what it is doing (e.g., tracking a moving ground vehicle) and task the remaining two UAVs to monitor the three NAIs (including some swapping of camera foci).
5. *Prosecute Target*—Given that a vehicle is already being tracked, task the coordination of lasing and firing at that target (after a final human authorization).

Initial studies using this set of plays in MUSIM were conducted by Fern and Shively [3] and showed significant advantages for play usage over the traditional waypoint-based flight controls representative of current operational UAVs. Fern and Shively's goal was to compare this more manual "tools" control mode vs. tools + single ship plays (1 and 2 above) vs. tools + single and multiple ship plays (1-5). Dependent variables were performance on the primary task (detecting and tracking military vehicles and prosecuting those which were weaponized) as well as on a secondary task ("painting" civilian vehicles via a designator button press). Results showed improved accuracy, reduced reaction time and reduced workload (by TLX ratings) for play usage—and these values were significant for the multi-ship plays on the secondary tasks (and

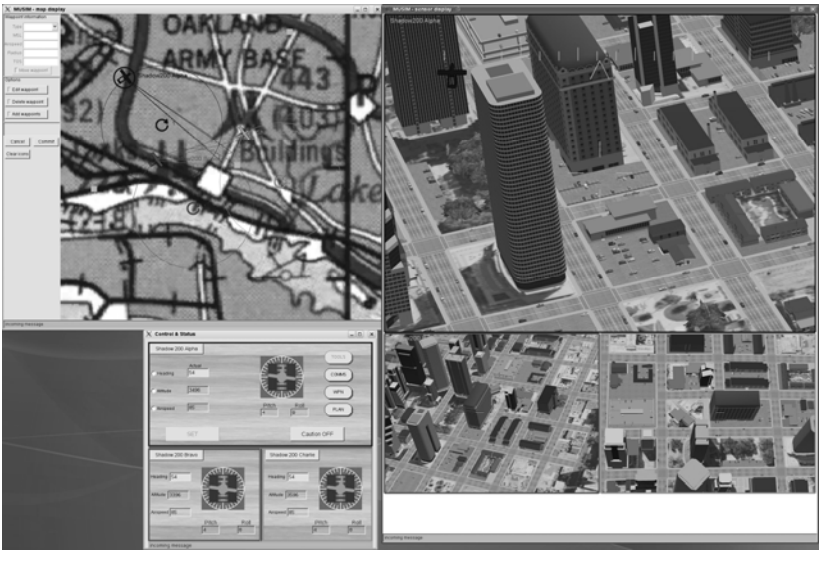

Fig. 2. The MUSIM simulation environment

in a similar direction, but not significant for primary tasks). Our own work began with a similar study varying operational tempo (through variations in the number of ground vehicles present) which showed similar results: in both high and low tempo circumstances, operators painted significantly more civilian vehicles, and tracked and prosecuted more military vehicles (non-significantly) when they had access to plays.

But again, the plays were designed to support exactly these tasks in this environment. In a subsequent pair of studies, we have included circumstances for which the set of plays is non-optimal. This has been accomplished by inserting a "popup" threat which is not in a pre-designated NAI location, and is located outside the geographic area in which the baseline tasks described above are conducted. Subjects are notified of this threat and the (high priority) need to prosecute it by means of an incoming text message. Since this popup occurs in an "undesignated" location, the play-based software the subjects use doesn't "know" about it—which means none of the plays above can be called for it (i.e., "Monitor NAI" won't work since it is not in a designated NAI location). Instead, subjects in all conditions have to use the "manual" level of tools to create a waypoint path for a UAV to go and find the popup threat, and then must manually lock on to it, lase it, hand off coordinates and prosecute it. Of primary interest was whether subjects who had had use to plays throughout a longer period of operating in the MUSIM environment would be at a disadvantage (either overall or specifically in dealing with the popup threat) relative to subjects who had been forced to use the more basic, waypoint-based UAV control tools throughout.

4.1 Experiment 1—Short Trials, Moderate NOPE Frequency

In our first NOPE experiment (presented in more detail in [9]), we were also interested in varying operational tempo (and therefore taskload) levels. Each of 15 participants completed 6 missions, each one including a single NOPE event. The experiment used a 2 × 3 within-subjects design with two levels of operational tempo/taskload (low, high) and three levels of control mode (Manual Tools, Single Ship Plays + Tools, and Single Ship + Multi Ship Plays + Tools). In low tempo conditions, 19 civilian vehicles were presented; in high tempo conditions, 64 civilian vehicles were presented (frequencies both higher and lower than those used in [3]). All trials lasted 10 minutes. Subjects were trained, demonstrated proficiency on the range of control modes available, and then performed their six trials in a randomized order.

Results were evaluated on the accuracy and speed of performance on the ongoing tasks (target tracking and civilian vehicles painting), as well as the ability to find and destroy the popup threat. Prosecuting weaponized vehicles occurred very infrequently and was omitted from the analysis. As with the prior studies described above, subjects generally performed better on the "normal" (i.e., non-NOPE) tasks when they had the full range of plays available. Tracking accuracy showed that participants established track on significantly ($F(2, 28)= 4.3$, $p < .05$) more military ground vehicles when they had the full range of plays available (M = 61.7, SE = 8.6) than when they only had manual tools (M= 46.1, SE = 6.3), with single ship plays + tools falling between those extremes ($M = 49.4$, $SE = 6.2$). For the secondary task of painting civilian vehicles, access to tools provided small advantages in accuracy and reaction time, but these did not reach significance. High operational tempo hurt performance (significantly

reducing accuracy and increasing reaction time), but there was no significant interaction of tempo and control mode—indicating, that a range of plays provided benefits that were not sensitive to workload levels, at least within the ranges tested.

But performance in the NOPE events was of more interest. If an "automation complacency" effect [8] was produced by consistent use of optimal plays before the NOPE, then we expected subjects to perform worse during NOPE in trials where they had access to play-based control vs. using manual tools throughout. But this is not at all what occurred. Instead, participants were significantly ($F(2,28)= 13.48$, $p < .05$—see Table 1) faster in dealing with the popup threat when they had the full range of plays available than with manual tools alone—even though those plays were no help to them in prosecuting the popup threat. Again, the single ship plays fell between the other control conditions. Operational tempo also produced a main effect ($F(1, 14) = 11.45$, $p < .05$) such that subjects were faster to prosecute the popup in low tempo conditions, but again there was no significant interaction between tempo and control.

Thus, far from finding evidence for a complacency effect with play usage, we found the opposite. Even though plays were not helping to perform the NOPE event, having them available during the remainder of the trial helped even during NOPE. Instead of producing over-reliance on plays, or loss of ability and familiarity with manual tools, having well-fitting plays during other portions of the trial may have freed up enough cognitive workload and situation awareness capacity to allow users to "stay ahead" of the situation and better deal with the NOPE when it occurred.

4.2 Experiment 2—Longer Trials, Rare NOPEs, Sequence Variations

It might reasonably be objected that having six ten-minute trials, each containing a ~3 minute NOPE event, hardly gave time for participants to develop "complacency" in automation use. Thus, in a second experiment, we made a more rigorous attempt to induce complacency effects. Trials were 30 minutes long and contained two NOPEs. The single-ship plays control mode was dropped and we compared only manual tools vs. full range of plays (single and multi-ship). Similarly, operational tempo was not included as a variable and an intermediate tempo was used. Finally, since prior work [10] showed that when a failure is experienced (early in one's work with automation vs. after a period of reliable performance) affects trust and usage decisions, we also explored this variable by contrasting trials in which the NOPE events happened close to each other near the end of the 30 minutes vs. others in which one NOPE event happened within the first 5 minutes and the second happened at ~25 minutes into the trial. We called these the Late/Late (or L/L) vs. Early/Late (E/L) sequence conditions.

Thus, experiment 2 was a 2x2 blocked design with NOPE timing (E/L vs. L/L) being one factor and control mode (Tools vs. Plays) being the other. Each subject received two trials instead of the four required for a full between-subjects design. The different blocked combinations

Table 1. Time (in sec.) required to prosecute the NOPE event with different control modes

	M	SE
Manual Tools	214.22	6.19
Single Ship Plays	189.49	6.00
Single & Multi-Ship Plays	175.11	3.64

of trials were: (1) E/L + Tools, then L/L + Plays, (2) E/L + Plays, then L/L + Tools, (3) L/L + Plays, then E/L + Tools, (4) L/L + Tools, then E/L + Plays. Twenty subjects were each randomly assigned to one of the blocks.

The consistent findings from prior studies of advantages for plays on the non-NOPE tasks were largely absent here. We saw no significant effects of control mode over the full 30 minute trials. This may have been due to the added time available for participants to become familiar with the tools control mode—allowing increased competency with the more difficult manual controls to produce a ceiling effect.

There were, however, interesting findings in performance on the popup. If a complacency effect for plays exists, we expected prosecuting the popup to be slower for participants in play conditions vs. tools. This effect was expected to be larger for those who had a longer period to become complacent (those in the L/L condition).

Again, this is not what was observed (cf. Fig. 3). There was a main effect of popup sequence (F (1, 19) = 6.13, $p < .05$) with the second popup in each trial being prosecuted faster than the first. There was also a significant interaction of popup sequence with timing (F(1, 19)= 14.66, $p < .01$) such that participants were much slower to prosecute the first popup in E/L trials than in L/L trials. Fig. 3 makes it clear that this was largely due to a much slower response from subjects in the plays/tools condition. Looking only at data for E/L trials, we see that the second popup was prosecuted faster than the first (F (1, 16) = 13.20, $p < .01$) and that tools control alone was marginally faster (F(1, 16) = 3.69, p= .07) than control via plays + tools. The interaction between Popup position x Control mode was non-significant (F(1,16) = 2.23, p = .155) in spite of the large apparent Plays decrement for the first popup.

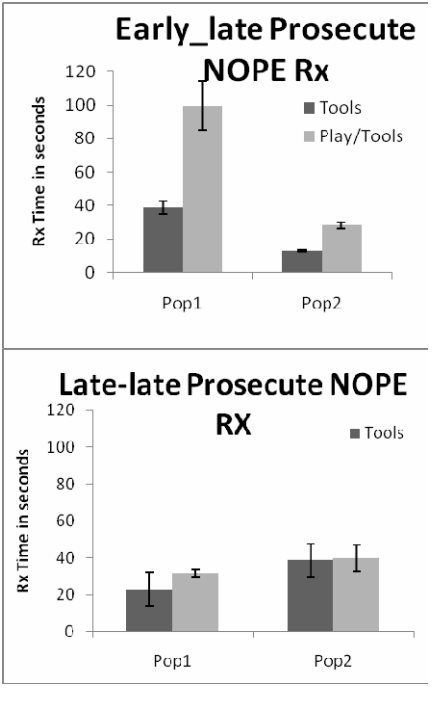

Fig. 3. Reaction time results for prosecuting the NOPE in Experiment 2

But note that this is not at all what we would have expected if the use of plays in optimal conditions produced complacency and poor performance in suboptimal conditions. If that had been occurring, we should have seen greater complacency the longer subjects had to experience the optimal use of plays—in the L/L condition. Instead, participants using plays are more disrupted in the first (earliest) NOPE event they encounter. While this may be evidence of overreliance on plays, it would appear that that overreliance decreases over time, rather than increases.

5 Discussion and Conclusions

Delegation control as a model for interaction with automation (and, in particular, unmanned vehicles) is increasingly showing promise, but it is unlikely to be a panacea. After all, human-human delegation and "supervisory control" is far from perfect. We went looking for complacency and over-reliance on plays under conditions where they were not optimal, but in spite of allowing subjects up to 30 minutes to work with plays and almost 25 minutes before providing a non-optimal event, we saw no consistent evidence for such effects. The finding that plays were significantly worse in handling the *first* popup in the E/L condition in Experiment 2 is exactly the opposite of a complacency interpretation. While this may point to a need for added training to become fully comfortable with the use of plays and tools concurrently, it provides no support for the claim that plays lead to unique decrements in some circumstances.

Acknowledgments. This work was funded by a SBIR grant from the U.S. Army Aeroflightdynamics Directorate, contract # W911W6-08-C-0066. The authors would like to thank Jay Shively, Lisa Fern and Susan Flaherty for both management oversight and significant technical contributions which lay the groundwork for this work.

References

[1] Miller, C.A., Parasuraman, R.: Designing for flexible interaction between humans and automation:Delegation interfaces for supervisory control. Human Factors 49, 57–75 (2007)

[2] Parasuraman, R., Galster, S., Squire, P., Furukawa, H., Miller, C.: A flexible delegation interface enhances system performance in human supervision of multiple autonomous robots: Empirical studies with RoboFlag. IEEE Trans. Systems, Man, & Cybernetics 35, 481–493 (2005)

[3] Fern, L., Shively, R.J.: A comparison of varying levels of automation on the supervisory control of multiple UASs. In: Proc. AUVSI's Unmanned Systems, North America, Washington, D.C. (2009)

[4] Kirwan, B., Ainsworth, L.: A Guide to Task Analysis. Taylor & Francis, London (1992)

[5] Nau, D., Au, T., Ilgami, O., Kuter, U., Muñoz, H., Murdock, W., Wu, D., Yaman, F.: Applications of SHOP and SHOP2. Technical Report, University of Maryland (2004)

[6] Ingrand, F., Georgeff, M., Rao, A.: An Architecture for Real-Time Reasoning and System Control. IEEE Expert, 34–44 (December 1992)

[7] Parasuraman, R., Riley, V.: Humans and Automation: Use, Misuse, Disuse, Abuse. Human Factors 39(2), 230–253 (1997)

[8] Parasuraman, R., Molloy, R., Singh, I.: Performance consequences of automation-induced complacency. Int. Journal of Aviation Psychology 3, 1–23 (1993)

[9] Shaw, T., Emfield, A., Garcia, A., de Visser, E., Miller, C., Parasuraman, R., Fern, L.: Evaluating the Benefits and Potential Costs of Automation Delegation for Supervisory Control of Multiple UAVs. In: 54th Meeting of the Human Factors and Ergonomics Society, pp. 1498–1502. HFES Press, Santa Monica (2010)

[10] Rovira, E., McGarry, K., Parasuraman, R.: Effects of imperfect automation on decision making in a simulated command and control task. Human Factors 49, 76–87 (2007)