Delegation and Transparency: Coordinating Interactions So Information Exchange Is No Surprise

Christopher A. Miller

Smart Information Flow Technologies (SIFT), 211 First St. N. #300,
Minneapolis, MN USA 55401
cmiller@sift.net

Abstract. We argue that the concept and goal of "transparency" in human-automation interactions does not make sense as naively formulated; humans *cannot* be aware of everything automation is doing and why in most circumstances if there is to be any cognitive workload savings. Instead, we argue, a concept of transparency based on and shaped by delegation interactions provides a framework for what should be communicated in "transparent" interactions and facilitates that communication and comprehension. Some examples are provided from recent work in developing delegation systems.

Keywords: flexible automation, adaptive/adaptable automation, Playbook®, delegation, Uninhabited Aerial Systems, trust, transparency, supervisory control.

1 Introduction

"Transparency" has been held up as a goal for automated systems that assume or require a substantial human interaction ([1], though the term is not used in this paper), and on its surface this seems a laudable and reasonable goal. But what does transparency in human-automation interaction mean anyway? What can or should it mean?

Naively, it would seem that "transparency" is a straightforward property such that all a system's functions and behaviors, as well as the rationale behind them, are available and obvious to human users. The automated system is "like glass" in that its workings are apparent to all. Achieving such transparency might be a substantial challenge for user interface designers, and understanding all the behaviors and rationales transparently presented might require specialized knowledge (e.g., aeronautical mechanics and flight controls for an autopilot), but these do not seem to be impossible goals, at least in principle.

The problem is that this form of transparency is fundamentally at odds with the goals of most human-automation interaction in the first place. Automation is generally created and deployed to save the human operator effort and/or to achieve performance speed or accuracy beyond what a human alone could do. This means that if transparency demands that the human maintain awareness of every sensed observation, decision and executed behavior the automation performs, then there is no time or

R. Shumaker and S. Lackey (Eds.): VAMR 2014, Part I, LNCS 8525, pp. 191–202, 2014.

(cognitive) effort saved. Full transparency in automation (and the human responsibility to make full use of it) would eliminate most benefit from the automation[1].

2 The Problem of Transparency

In earlier work [2], we formulated a concept of the relationship and tradeoff between human cognitive load, unpredictability to the human in automation behaviors and the achieved competency of the overall human-automation system (that is, performing correct behaviors in context). See Figure 1. Very frequently, we implement automation with the objective of achieving greater competency (the ability to achieve more behaviors more accurately, precisely or rapidly, etc.)—this corresponds to expanding the length of the triangle base. The relationship to the other legs of the triangle, however, illustrates two additional principles. First, that an increase in competency can only be achieved via an expansion of one or the other (or both) of the other two dimensions. Second, that a given level of competency can be achieved in a variety of ways that will differ in impact on workload and unpredictability.

Competency is achieved by giving the responsibility for monitoring, assessing and making decisions and executing actions to some agent—either human or machine. If the added responsibility (and the corresponding added tasks) for performing these steps are all given to the human, then clearly, added workload will be the result. If those added responsibilities are given to automation, the human cognitive load will not increase, but the knowledge, awareness and control which comes from performing those tasks will also not accrue to the human. If the human is required to maintain awareness of everything the automation is sensing, assessing and deciding, as in a fully "transparent" system, then awareness will not decrease, but the human will have performed at least all of the cognitive work that s/he would have had to perform to achieve the task in the first place. In short, for competency increases without cognitive workload increases, it is inevitable that some of the sensing, assessing

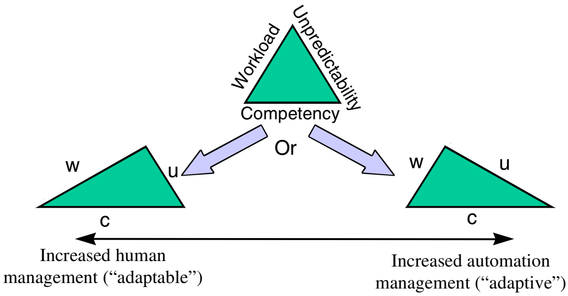

Fig. 1. The spectrum of tradeoffs between competency, workload and unpredictability (from [2], used with permission)

[1] There are exceptions to this general formulation. Some automation enables performance at times or places where humans cannot. It might well be acceptable to use "transparent" automation in the sense described above in situations hostile to human presence, but where sufficient time and human resources are available to fully understand everything the automation is doing—such as in Martian rovers and nuclear reactor maintenance.

and deciding activities be taken out of the human's hands—that is, they must be "obscured, not "transparent" in the sense above if any cognitive work is to be saved.

We have not previously considered the meaning of the height of the triangle, but it seems reasonable that it represents an abstract measure of work complexity: the amount of cognitive work which *must be done by someone* (humans or automation), for the level of competency. Work complexity is probably not as independent of the competency dimension as implied by the triangle figure—a point at which the analogy breaks down. Work may be made less complex by better system design, reduction in the number of interacting components, etc.—and such reductions can reduce both human cognitive workload and system work producing unpredictability. For example, by most accounts, jet engines are less complex than traditional piston driven turbines because there are fewer moving parts. This implies that to be fully aware of the state and behaviors for operating a jet engine is "simpler" than for a piston-driven turbine, whether it is a human or automation doing it.

Transparency in a human-automation system is essentially the opposite of the unpredictability leg of the triangle in Figure 1 and, therefore, "transparent design" would imply striving to minimize that leg. As for competency though, for a given level of work complexity, this can be accomplished only by reducing either human cognitive workload or the competency of the overall human-automation system, or both. If the goal is to preserve overall system competency and if no reduction in the underlying work complexity is possible, then the only way to accomplish that while concurrently reducing unpredictability is by increasing human cognitive workload.

A pair of objections to this reasoning seems valid. First, this reasoning only applies to cognitive workload. Some automation performs mainly or solely physical tasks. Reducing unpredictability through increased transparency for such a system would still increase the human's cognitive load, but there could be substantial savings in physical workload. Second, one might object that "transparency" does not refer to the human's need to be aware of all operations of the automation concurrently, but only to the availability of that information. Automation should be transparent, but the human has to decide when and what to look at. I would agree that removing the need that the human maintain awareness of everything avoids the problems above, but relaxing that requirement begs the question of how to design, select and train the human-automation system to afford the right kind of awareness for good performance and safety. We will advance some thoughts on that in the next section below.

3 Practical Transparency—The Role of Delegation

So if "transparency" cannot, practically speaking, mean that the human knows everything about what the automation is doing, then what can or should it mean? Chen, et al., [3] define automation transparency as "… the descriptive quality of an interface pertaining to its abilities to afford an operator's comprehension about an intelligent agent's intent, performance, future plans, and reasoning process." If this cannot reasonably mean full awareness of all these elements, then the emphasis is on "operator's comprehension" and the key question becomes how much and what type of awareness is necessary to promote comprehension in a multi-agent system?

This question requires a decision about the roles and relationships of the system's actors. Most current and near-future visions of human-automation interactions leave the human in charge of automation in a *supervisory control* relationship [4]—that is, both responsible for directing the automation and for ensuring all functions are accomplished. This relationship demands more awareness of a greater range of functions than other possible ones, and it is subject to the fundamental limitation described above: the human cannot be aware of everything the system is doing if any cognitive workload is to be saved. Humans in supervisory positions exert control through *delegation*—the act of giving instructions or orders that the subordinate is expected to attempt to follow and perform, with some reporting throughout execution and discussion when compliance with the directives are difficult, impossible or sub-optimal.

In multiple efforts, we have explored enabling humans to express intent and delegate to subordinate automation with the same flexibility possible in human supervisory control [2,5]. We use the metaphor of a sports team's playbook. Our Playbook® systems allow humans and automation to share a conceptual, hierarchically-structured framework for the goals and methods of a pattern of activity (a "play") and to delegate instructions and discuss performance within that framework.

Flexible delegation achieves *adaptable*, rather than adaptive, automation [2]. To achieve a greater range of competency from automation than available in traditional, static automation, researchers turned to adaptive automation approaches [6], which exhibit a wider variety of behaviors but which leave decisions about when and how to shift behaviors to the automation itself. By contrast, in adaptable automation, the human initiates behaviors by "delegating" at flexible levels of specificity; automation is then responsible for planning and executing within the delegated instructions. Adaptive automation is targeted at saving the user workload and may result in superior performance in some contexts, but when the user and automation are at odds as to what should be done, the human has little opportunity to influence, override or even understand the automation and may end up "fighting" it for control.

We have argued for adaptable approaches [2] in most contexts due to this potential for mismatch. Adaptable, delegation approaches have been shown to result in improved overall system performance when examined across unpredictable and/or unexpected contexts [7] and reduced human workload relative to adaptive systems in some circumstances [8]. There is also reason to believe that the act of expressing intent is an important part of the naturalistic decision making process [9], serving to "crystalize" intent for both the declarer and the hearers. Moreover, the process of declaring intent to subordinates should facilitate situation awareness of what the subordinate is doing (alleviating the 'what is it doing now' problem [1]) and even, potentially, improving trust accuracy by providing the "truster" with an explicit, declared intent to evaluate the "trustee's" performance against. Many of these effects have been observed for adaptable automation approaches in recent studies [7,8,10,11].

The question considered below is whether intent declaration and intent-focused interactions inherent in delegation systems may have an impact on "transparency" and transparent design. Since the key is to convey information which will "afford operator comprehension" as appropriate to the operator's role as the supervisor in a supervisory control system, delegation informs the behavior expected of the subordinate.

3.1 What Is Delegation?

Collaborative interactions in work domains are primarily about intent—to perform an action, use a resource—and the need or desire to notify others, receive permission, elicit cooperation, report status, etc. against it. When the operator is a supervisor, these interactions become instructions (as Sheridan defines for supervisory control [4]) with an expectation of compliance. This is delegation.

Intent may be expressed in one of five ways (cf. Table 1). The supervisor may express a goal (a world state) to be achieved or a plan (a series of actions) to be performed. Constraints and stipulations on actions, methods or resources to be used may also be expressed. Finally, less specifically, the supervisor may also express values or priorities. These refer to the relative goodness or badness of states, actions, resource usages, etc. if they are achieved or used. These methods are rarely mutually exclusive and may be combined to achieve various methods of delegation as appropriate to the domain, and the capabilities of both the supervisor and subordinates.

Delegation is inherently hierarchical. Goals and tasks are composed hierarchically in a causal means-ends fashion—as expressed in traditional task analysis techniques [12]. Our Playbook implementations have used this structure to facilitate optional operator input in a fashion that enables AI planners to create plans adhering to both the shared play definition and to any additional stipulations the operator provides [2]. Even delegation interactions which center on resource usage also participate in hierarchical decompositions along part-whole dimensions—such that resources are usually parts of larger wholes, and may involve decompositions into smaller sub-parts.

Table 1. Five methods of intent expression in delegation

Supervisor Method	Subordinate Responsibility
Goal	Achieve goal if possible; report if incapable
Plan	Follow plan if possible; report if incapable
Constraint	Avoid actions/states if possible; report if not
Stipulation	Achieve actions/states if possible; report if not
Value Statement	Work to optimize value

Thus an act of delegation expresses the operator's intent for a constrained, but still under-specified, set of behaviors, expressed either implicitly or explicitly, to be accomplished by one or more subordinates. This expression of delegated intent, we argue, frames the interaction and helps to determine the kinds of information which will "afford operator comprehension" in a transparent system—in ways discussed below.

3.2 Delegation, Situation Awareness and Trust—The "Intent Frame" Effect

We argued, in [2], that any workload savings from efficient automation design could be devoted to maintaining or achieving better overall situation awareness about the context of use. While true, this phenomenon is partially countered by the unpredictability effect (described above) resulting from giving tasks to subordinates who, almost inevitably, must also be accorded some autonomy in their performance.

Situation Awareness (SA), as traditionally defined and measured [13], refers to all situational knowledge required to perform one's job. Knowledge specifically about what a subordinate is doing and why is surely part of that set, but it is more specific and subject to different influences. SA reduction specifically about delegated tasks is tolerated, even embraced, in exchange for competency improvements and/or workload reductions as discussed earlier. That said, the reduction in automation-related SA can differ in different human-automation interactions. In adaptive or traditional automation, behaviors are disconnected from human intent and therefore, an additional element of unpredictability enters into the human's experience. This is summed up in Sarter, Woods & Billings [1] work on "automation surprises"—instances in which non-transparent automation does unexpected, difficult-to-explain things. While transparency can alleviate automation surprises by giving humans insight into what the automation is doing, this comes at the expense of cognitive workload—since it requires the human to monitor and interpret those interfaces. This is akin to having a subordinate who one has to watch all the time to make sure what s/he is doing is appropriate.

Delegation provides another way. By tasking the subordinate, the act of delegation provides an *Intent Frame* that expresses and defines expectations about the subordinate's behavior for both parties. In communication, this frame explicitly details what the supervisor expects the subordinate to do, therefore that aspect of SA should improve for both parties. But delegation also helps awareness and interpretation of observed subordinate behaviors as well because it creates a cognitive expectation about what the subordinate is *supposed* to do. If a task (or "play") is delegated, then certain behaviors are expected and others are not. This framing narrows the set of behaviors that need be attended to by the superior: instead of checking to see what behaviors, from all possible ones, the subordinate is doing, s/he may simply check to see whether the subordinate is doing what was expected or not. Even unanticipated behaviors can be interpreted more directly for whether they are reasonable within the intent instead of for what they could possibly accomplish. In short, explicitly delegated intent shifts the operator's task in monitoring and interpretating automation from one of "what is it doing now?" to a cognitively simpler one of "is it doing what I told it to do?"

The same Intent Frame effect likely has an impact on trust formation. Lee and See [14] define trust as "…the attitude that an agent will help achieve an individual's goals in a situation characterized by uncertainty..." (p. 51). Thus, delegation impacts trust by making it clear(er) to both parties what those goals are. While this may or may not lead to increased trust depending on the subordinate's behaviors, it does "sharpen" the test for trust and should, therefore, speed trust tuning [cf. 15].

These hypothesized effects on trust and automation-specific SA have largely not been tested directly, but there is some indirect support for them. Several studies [8,11] report improved performance and/or faster response times on secondary tasks when using adaptable delegation—which might imply either improved SA or reduced workload or both, though the former was not explicitly measured. In [8], experimenters conducted a direct comparison of adaptive vs. adaptable automation on tasks representative of multi-UAS operations, and reported higher subjective confidence ratings (a loose analog for trust) under adaptable vs. adaptive control. Finally, Layton, Smith and McCoy [16] had pilots interact with three kinds of automated route planning support in a commercial aviation context: a “low” automation level where operators sketched routes and automation computed route details such as fuel consumption and arrival times, a “high” level providing expert system-like support proposing a single complete route to the pilot, and an “intermediate” level where the pilot had to request a route with specific constraints (e.g., ‘going to Kansas City and avoiding Level 3 turbulence’) before automation developed it. Pilots with the intermediate and high automation levels explored more routes because manual sketching was too difficult to allow much exploration, but with full automation, users tended to accept the first route suggested without exploring it or alternatives deeply. Particularly in trials where automation performed suboptimally (e.g., failing to consider uncertainty in weather predictions), humans using the intermediate level produced better overall solutions. Although SA was not directly assessed, this suggests that pilots were most able to bring their own knowledge to bear, and most aware of the automation’s plans when they explicitly instructed it as to what they needed.

3.3 Delegation and Dialog Framing—Improvements in Communication

While the prototypic delegation interaction is the supervisor conveying intent to the subordinate, other interactions flow from both parties. Delegation dialogs, and the hierarchical structures which underlie them, also serve to frame and facilitate communication from automation to the supervisor in a variety of ways and are therefore relevant to transparency, as will be discussed for a suite of different technologies below.

“Explanation” and Negotiation through Relaxed Constraint Planning. “Transparency” in human interactions is greatly facilitated by natural language explanations, but human-understandable explanation of the complex reasoning of a mathematical control or symbolic logic system has been a canonically difficult problem in Artificial Intelligence for decades [17]. Furthermore, explanation is a key to effective, multi-dimensional negotiation since it facilitates understanding of the parties’ goals and identification of potential tradeoffs they may be willing to make.

In one Playbook® implementation, we provided a simple explanation and negotiation approach integrated with our automated planner [18]. Prior Playbook versions had simply tried to create an executable plan given the supervisor’s instructions. If this failed, the system reported the failure but did not otherwise indicate *why* the user’s instructions were impossible. This is akin to a subordinate who, responding to instructions, says simply “I can’t”—a non-transparent (and unhelpful) behavior. To improve, we implemented a “Relaxed Constraint Planning” (RCP) approach. Under RCP, when the system receives instructions it cannot achieve, instead of responding “I can’t” it seeks to progressively relax constraints until a valid plan can be provided. This now is akin to saying “I can’t do that, but here’s something close I can do. How’s this?” This has many benefits, including a form of increasing transparency.

RCP not only provides a valid plan that can be executed immediately, it also gives insight into what the subordinate system had to change to achieve performability. If, say, the system had to arrive later than requested, the user can decide whether relaxing a different constraint (say fuel consumption) is preferable. Thus, RCP can serve as the first "move" in a negotiation process. Constraints in our initial RCP system were relaxed via an ordered list of static priorities, but this could be both more dynamic and knowledge-based. Communication of multiple alternatives and/or visualization of the effects of alternate constraint dimensions would speed the "negotiation" process by increasing transparency to the system's underlying reasoning system. Importantly, however, framing this negotiation in a task-based delegation process keeps the goals and methods grounded in a shared understanding of possible approaches.

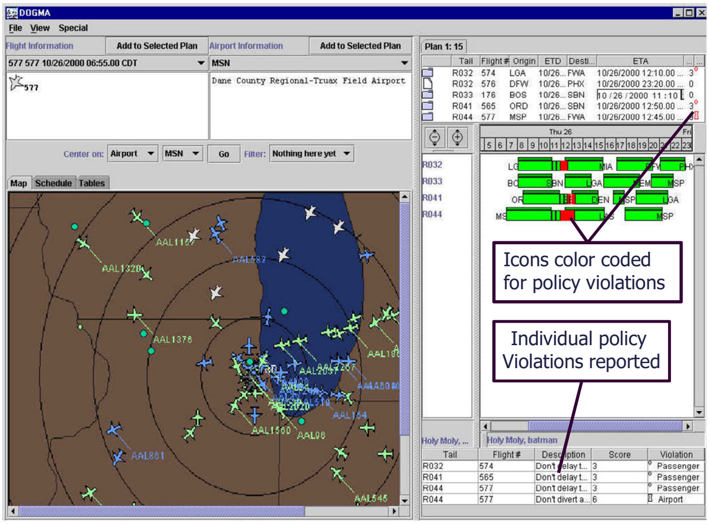

Fig. 2. Implemented policy visualization system

Policy Visualization. Particularly for policy and values (cf. Table 1), delegation offers another method to support explanation and negotiation. If a supervisor provides an explicit (and, ideally, quantified) set of policy statements about conditions or approaches with associated values, then plans and projected outcomes can be evaluated against them and the results shown in a variety of ways ranging. We used this approach in a prototype system developed for dispatchers in commercial airlines [19] who must make decisions about where to divert airplanes when they, unexpectedly, cannot make their destinations (e.g., due to weather events, etc.). These diversion decisions can affect many stakeholders in the organization, each with differing priorities. Worse, values change in different contexts—e.g., during holidays. Finally, dispatchers have ~5-15 minutes to make decisions that can affect many flights and it is exceedingly difficult to maintain awareness of the priorities of all stakeholders.

By capturing a numerically-weighted value for each state of concern and "bundling" these policies by stakeholder, we provided optimization functions that could serve to visualize the good or bad aspects of potential outcomes or, with a search algorithm, to generate criteria-optimized outcomes. Separating policy statements by context and by stakeholder allows the dispatcher to assert or weight different policy bundles. As plans are developed, they can be reviewed against values either separately (to determine effects on different stakeholders) or in aggregate. Policy bundles can be weighted differently to reflect permanent or temporary variations in the importance of different stakeholders. Figure 2 illustrates an implemented prototype, developed by Honeywell Laboratories, using this approach for visualizing alternate dispatch plans.

In short, policy value statements, especially if "bundled" by various concerns, enable rapid conveyance of how well alternate plans do against the values, how specific policy concerns are contributing to that value, which concerns are satisfied more or less, etc. Insofar as automation is creating or critiquing plans for a supervisor, such reports can go a long way toward transparently conveying the automation's reasoning.

Information Expectations from Delegation. Delegation and negotiation produce an agreed-upon plan that carries assumptions about what should be communicated back to the supervisor—under normal and abnormal circumstances. These expectations do not promote full transparency so much as *principled information reduction* to satisfy Chen, et al's goal of "operator comprehension" while maintaining reasonable workload. This principle manifests in various ways, many of which have been explored as part of the Flexible Levels of Execution—Interface Technologies (FLEX-IT) project [4]. FLEX-IT is developing an adaptable delegation approach to controlling multiple UASs via highly flexible automation interactions—from manual flight control to high level, multi-vehicle play calling, all via multiple interface modalities.

First and most obvious, as part of the act of delegation, the supervisor may explicitly include instructions about reporting. FLEX-IT uses versions of this approach via incorporating "decision" and "notification points" in plays. These are "points" tied to absolute or route-based geographic locations (or, potentially, to temporal- or event-based "points") at which the subordinate is to report to the supervisor (for notifications) or ask for further instructions (decisions). In essence, they permit the supervisor to say "Do X, and when or if Y happens, let me know [ask me what to do next]." Figure 3 illustrates a notification point from FLEX-IT.

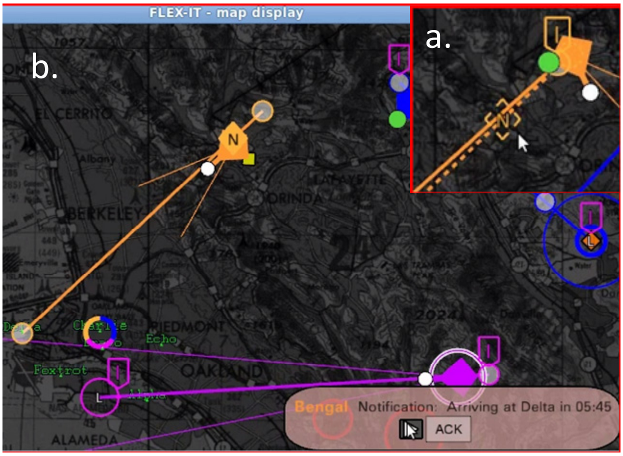

Fig. 3. Notification point behaviors showing (a) insertion of a point, and (b) activation of the point when reached by the UAS

Beyond explicit instructions, FLEX-IT also uses delegated activities as a "vocabulary" to

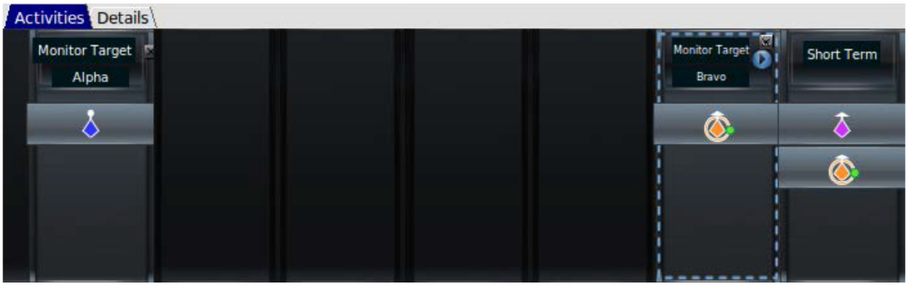

Fig. 4. Activities Panel showing vehicles allotted to specific tasks and under "Short Term" control (available)

facilitate and organize information exchange. FLEX-IT initially organized information and status presentations around the individual vehicles being controlled—a typical format for Air Force displays—but it became clear with multiple vehicles involved in different tasks that organization around the task itself was at least as useful. Thus, in addition to the typical map display, we added an "Activity Panel" which organized vehicle icons by the activity they were engaged in, presented task-relevant status of those vehicles (e.g., whether they were currently in position to transmit imagery for a monitoring task), provided links to more detailed information about the task and, potentially, to temporal views of task performance, etc. (See Figure 4). The human or automation could move vehicles from task to task within the Activity Panel. The motion itself was salient, helping to mitigate change blindness, thereby maintaining SA about who was doing what. Vehicle tasks were also reflected by icons and glyphs (e.g., a shared, multi-colored ring icon to reflect vehicle monitoring ranges in a shared monitoring task) and could be extended to reflect task status (e.g., on time or not). Finally, the vocabulary of task labels also served multiple purposes in the design of multi-modal interactions—allowing operators to designate a group of UASs to halt or modify their stealth profile by referencing their shared task or by using the sub-task decomposition as an underlying structure for sketch and speech interactions (i.e., "Ingress like this [sketch route], then monitor here [touch] using this pattern [sketch]").

A final use of the delegation interaction to manage information flow is more subtle and largely untried to date. We have created dynamic information management systems in the past (e.g., [20]) which were essentially adaptive automation for an operator's displays. Our approach has been to represent information needs abstractly for the types of tasks an operator is likely to perform and then, either through inference or explicit declaration, assess the tasks that are actually being performed from moment to moment throughout the mission. Available displays are then configured to supply as much as possible of the set of information needs for those tasks. A similar approach could be formulated for adaptable automation by developing heuristics for how to interpret delegated instructions in terms of their information reporting requirements. For example, it seems reasonable that supervisors will want to be informed of task completion and of any circumstance that makes task completion impossible. Resource consumption (including time) is probably not important information to report if it is proceeding as expected in the agreed-upon plan. For example, a vehicle performing a steep dive is not overly noteworthy if the plan included that action, but is highly noteworthy if it did not. When deviations are expected to exceed a pre-defined threshold, however, then reporting becomes important. For specific resources (e.g., a vehicle that the supervisor is holding in reserve, entering a restricted airspace) or specific actions (e.g., firing a munition, decreasing stealth by descending below a specified altitude), notification might always be required. We have not worked out this method completely, but we believe a reasonable and general approach to information reporting could be built around delegated task heuristics such as these.

4 Conclusions

In the multiple examples provided above, we have illustrated how taking a "delegation perspective" on the concept of transparency may lead us out of the impossible and counterproductive attempt to achieve "full transparency" and toward a more productive goal of conveying specifically what is necessary to "afford operator comprehension" of the behavior of a subordinate. Furthermore, we have argued for how task-based delegation can frame the human-automation interaction both to improve communication to and comprehension of the automated subordinate, but also to improve and tune communication, expectations and interpretations for the supervisor. It does this by establishing an "Intent Frame" between the supervisor and subordinate (and, at least as important, within the mind of the supervisor) which serves to restrict the space of what must be communicated and shape and speed the interpretation of that which is communicated. While much of what is argued above remains to be proven, it seems to flow reasonably from the nature of delegation interactions in both human-human and human-automation interaction.

Acknowledgments. The thoughts expressed here are the work of the author, but related work has been supported and augmented over many years by many individuals. Particular thanks are due to Drs. Mark Draper and Gloria Calhoun of AFRL/RHCI, Heath Ruff and Tim Barry of Ball Aerospace & Technologies, Dr. Jay Shively and Ms. Lisa Fern of the U.S. Army AeroFlightDynamics Directorate, Drs. Raja Parasuraman and Tyler Shaw of George Mason University, Dr. Michael Dorneich of Honeywell and Mr. Josh Hamell of SIFT. The FLEX-IT project was funded by AF/A2Q ISR Innovations Division and managed by at Wright-Patterson Air Force Base under Air Force Contract FA8650-08-D-6801. The Policy Visualization system was supported by funding from Honeywell Laboratories.

References

1. Sarter, N.B., Woods, D.D., Billings, C.E.: Automation Surprises. Hndbk. of HFES 2, 1926–1943 (1997)
2. Miller, C., Parasuraman, R.: Designing for Flexible Interaction Between Humans & Automation: Delegation Interfaces for Supervisory Control. Human Factors 40(1), 57–75 (2007)

3. Chen, J.Y.C., Boyce, M., Wright, J., Procci, K., Barnes, M.: SA-based Agent Transparency. ARL Technical Report (in prep.)
4. Sheridan, T.: Humans and Automation. Human Factors and Ergonomics Society, Santa Monica (2002)
5. Miller, C.A., Draper, M., Hamell, J.D., Calhoun, G., Barry, T., Ruff, H.: Enabling Dynamic Delegation Interactions with Multiple Unmanned Vehicles; Flexibility from Top to Bottom. In: Harris, D. (ed.) EPCE 2013, Part II. LNCS, vol. 8020, pp. 282–291. Springer, Heidelberg (2013)
6. Kaber, D.B., Riley, J.M., Tan, K.W., Endsley, M.R.: On the Design of Adaptive Automation for Complex Systems. Int'l J. of Cog. Ergo. 5(1), 37–57 (2001)
7. Parasuraman, R., Galster, S., Squire, P., Furukawa, H., Miller, C.: A Flexible Delegation-Type Interface Enhances System Performance in Human Supervision of Multiple Robots: Empirical Studies with RoboFlag. IEEE Sys., Man & Cybernetics—Part A 35(4), 481–493 (2005)
8. Kidwell, B., Calhoun, G., Ruff, H., Parasuraman, R.: Adaptable and Adaptive Automation for Supervisory Control of Multiple Autonomous Vehicles. In: 56th Annual Human Factors and Ergonomics Society Meeting, pp. 428–432. HFES Press, Santa Monica (2012)
9. Klein, G.: Naturalistic Decision Making. Human Factors 50(3), 456–460 (2008)
10. 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 Annual Human Factors and Ergonomics Society Meeting, pp. 1498–1502. HFES Press, Santa Monica (2010)
11. Miller, C.A., Shaw, T.H., Hamell, J.D., Emfield, A., Musliner, D.J., de Visser, E., Parasurman, R.: Delegation to Automation: Performance and Implications in Non-Optimal Situations. In: Harris, D. (ed.) Engin. Psychol. and Cog. Ergonomics, HCII 2011. LNCS, vol. 6781, pp. 322–331. Springer, Heidelberg (2011)
12. Diaper, D., Stanton, N. (eds.): The Handbook of Task Analysis for Human-Computer Interaction. Erlbaum, Mahweh (2004)
13. Endsley, M.R.: Measurement of Situation Awareness in Dynamic Systems. Human Factors 37(1), 65–84 (1995)
14. Lee, J.D., See, K.A.: Trust in Computer Technology: Designing for Appropriate Reliance. Human Factors 46(1), 50–80 (2004)
15. Miller, C.: Trust and Delegation: Achieving Robust Interactions with Subordinates. In: Burke, J., Wagner, A., Sofge, D., Lawless, W.F. (eds.) AAAI Spring Symposium on Robust Intelligence and Trust in Autonomous Systems. AAAI Press, Cambridge (in press)
16. Layton, C., Smith, P.J., McCoy, C.E.: Design of a Cooperative Problem-Solving System for En-Route Flight Planning: An Empirical Evaluation. Human Factors 36(1), 94–119 (1994)
17. Sørmo, F., Cassens, J., Aamodt, A.: Explanation in Case-Based Reasoning–Perspectives and Goals. Artificial Intelligence Review 24(2), 109–143 (2005)
18. Goldman, R., Miller, C., Wu, P., Funk, H., Meisner, J.: Optimizing to Satisfice: Using Optimization to Guide Users. In: American Helicopter Society International Specialists Meeting on Unmanned Rotorcraft. AHS Press, Alexandria (2005)
19. Dorneich, M.C., Whitlow, S.D., Miller, C.A., Allen, J.A.: A Superior Tool for Airline Operations. Ergonomics in Design 12(2), 18–23 (2004)
20. Miller, C.: Bridging the Information Transfer Gap: Measuring Goodness of Information "Fit". J. Visual Languages and Computing 10(5), 523–558 (1999)