Beyond Point Design: General Pattern to Specific Implementations

Joel Lachter[1(✉)], Summer L. Brandt[2], Garrett Sadler[3], and R. Jay Shively[1]

[1] NASA Ames Research Center, Moffett Field, CA 94035, USA
{Joel.Lachter,Robert.J.Shively}@nasa.gov
[2] NASA Ames Research Center, San José State University, Moffett Field, CA 94035, USA
Summer.L.Brandt@nasa.gov
[3] HATS Inc., 20944 Sherman Way, Suite 211, Canoga Park, CA 91303, USA
Garrett.Sadler@hats.solutions

Abstract. Elsewhere we have discussed a number of problems typical of highly automated systems and proposed tenets for addressing these problems based on Human-Autonomy Teaming (HAT) [1]. We have examined these principles in the context of aviation [2, 3]. Here we discuss the generality of these tenets by examining how they might be applied to photography and automotive navigation. While these domains are very different, we find application of our HAT tenets provides a number of opportunities for improving interaction between human operators and automation. We then illustrate how the generalities found across aviation, photography and navigation can be captured in a design pattern.

Keywords: Human-Autonomy Teaming · Automation · Human factors

1 Introduction

1.1 Problems with Highly Automated Systems

Elsewhere we have discussed a number of problems typical of highly automated systems [1]. Such systems are brittle, working properly within some bounded space for which they have been programmed, then failing when parameters fall outside that space. They are opaque, lacking transparency; human operators often do not know what the automation is doing or why. Operators often do not know when to trust automation, relying on it to handle conditions it cannot, or not taking advantage of it to handle conditions it can. As automation does more of the work, operators become less practiced. When the automation performs a task, the operator is often less aware of the system state.

While each of these issues is troubling by itself, they often manifest together. An operator, over-trusting the system, does not realize that some parameter has gone out of bounds. Because it is out of bounds, the automation either quits or is no longer reliable. The out of practice operator must then try to regain situation awareness using opaquely presented information. This situation has been responsible for a number of accidents (e.g., Air France 477, Korean Air 801). Similar concerns have been identified by others [4–6].

1.2 HAT Solutions

A number of authors have suggested that these concerns are ameliorated by developing interfaces and procedures analogous to guidelines for improving teamwork among humans [5, 7]. This area of research has been termed "Human-Autonomy Teaming" or "HAT." Our current approach to HAT emphasizes the following tenets:

Bi-directional Communication: Following the common aviation crew resource management (CRM) practices that encourage input from all relevant parties into decisions, communication should be bi-directional concerning all levels of planning and execution. At the highest level (the mission), this means that operators should be explicitly informing the automation of mission goals, and automation should be able to recognize when those goals are not being met and inform the operator (preferably with an alternate course of action). At the lowest level (implementation), the operator should be able to "see" what the automation is doing and understand why, propose adjustments and have the automation report the predicted consequences of those adjustments before execution. Similar bi-directional communication should occur at intermediate levels. To facilitate this dialog, automation should be able to present a rationale for recommendations and warnings, for example, indicating why a route is rated as unacceptable or what event triggered a warning light. In addition, under conditions of uncertainty, automation should indicate its confidence in the data or analysis being presented. This allows operators to better integrate information that may be significant but unreliable (e.g., when I need to leave for work might be significantly different if I want an 80% chance of being on time versus a 99% chance).

Transparency: Providing rationale and confidence levels also fall under a more general tenet, that automation should be transparent. Lyons [8] defines automation transparency as a shared awareness and shared intent between the system and its human operator(s). That is, the system and its operators should be able to recognize what the human/automated teammate is doing and why. To be truly transparent, communication should use a shared language: Automation should present information in ways that fit the operator's mental model.

Operator Directed Interface: Interfaces should allow for dynamic task allocation directed by the operator. In particular, intent inferencing about the operator's state or goals should be minimized.

1.3 The Value of a Generalizable Solution

Elsewhere we discuss our efforts to demonstrate and test these principles implemented in an airline dispatcher ground station designed for flight following [2, 3]. But to what degree can the success of that implementation be attributed to the HAT principles as opposed to other design considerations? One way to think about this problem is "How easy would it be to apply the HAT system we developed for our ground station to a novel situation?" This is the "turnaround test" discussed by Woods [9]. We are attempting to address this generalizability issue by first developing use cases that specify what HAT would look like in a broad variety of domains, and second extracting patterns from those use cases that capture generalities about HAT. In the following sections, we examine two such use cases and one such pattern.

2 HAT in Other Domains

2.1 Photography

Photography provides a very different set of use cases for automation and HAT from those associated with aviation. In photography the operator has much more latitude in setting goals and in many cases may have to change these goals quickly to capture fleeting images. Automation can help. It can focus and adjust exposure (the brightness of the picture), much more quickly and accurately than any human operator. However, first it must know the goal. Automation can also detect and react to items and events of interest such as faces, blinks, and camera shake. Thus the automation is capable of doing much of the work to realize the operator's goal for a shot, if that goal can be accurately conveyed. In this section we briefly discuss some of the choices facing a photographer and then discuss how these choices might be more quickly and accurately conveyed to the automation.

Perhaps the most obvious choice the photographer makes is what to focus on. Ignoring the issue of pointing the camera, there are typically many objects within the field of view that the operator could choose to be in focus. Objects closer or further than the chosen object will appear increasingly blurry in the final image. The issue here is that, while it is usually obvious to the operator what the picture is of, the autofocus system does not know. On many cameras, the autofocus system can be told to look for the closest object or faces, but those abilities are quite brittle, braking when there are multiple faces or the subject is not the closest object in the scene. Cameras come with these priorities because portraits and subjects in the foreground are common goals. But where does that leave someone taking pictures of flowers or waterfalls? It would be nice to have a more generalized solution.

Choosing what to focus on is, of course, not the only choice that must be made when taking a photograph. How bright (or dark) should the photograph be? How do we control the brightness? Without turning this paper into a Photography 101 textbook, let us just say that these are complex questions. Controlling the brightness of a photograph has side effects for how sharp the image appears; side effects that may be desirable, or not, depending on the goals of the photographer. While automation can determine the appropriate level of brightness quickly, different settings affect how it controls the brightness with major effects on the appearance of the final product.

We would like to mention one last type of automation that is creeping into some cameras. Cameras can read information off of the sensor and process it far more quickly than a human operator. As a result, it is possible to have the camera trigger (or briefly suppress triggering) based on certain events. Cameras can take a picture when lightening flashes, or, under flickering lights, at the brightest point of cycle. Cameras can momentarily delay firing when the subject blinks or when your hand is moving. This raises the possibility of having a camera fire at specific moments that the operator wants to capture, say, when a bat hits a ball.

Now imagine taking a hike with your son. You want different settings when you take a picture of your son, a flower, a waterfall, a fox that crosses your path. How can you easily move from one to another?

One solution would be to specify your goal, the kind of picture you want to take, at a high, "mission" level. Dialing in a Portrait, or Waterfall play would choose the appropriate settings. We envision the ability to create packages of settings offline that can then be loaded onto the camera. Following a similar proposal by Miller and Parasuraman [10], we refer to these packages as "plays." Plays could be "called" quickly by voice or using a scroll wheel. In some ways, plays may seem similar to the "scene modes" appropriate for many types of photography (sports, portrait, night, fireworks, etc.) offered on many cameras. However scene modes generally lack transparency, customizability (strategic, offline), and flexibility (tactical, real time). By allowing operators to create plays offline, it is possible to imagine building in crazy levels of specificity. Imagine a baseball play where you could specify the field location. If you then tell it you are taking pictures of the pitcher, and it senses your location (GPS) it can determine the approximate distance greatly improving autofocus speed and accuracy. You can specify the uniform color so it can select the appropriate people out of a scene. The operator could say "play at second" and the autofocus would focus at the appropriate distance and on the appropriate player, making it faster and less likely to focus on the wrong subject entirely. (Yes, one of us spends way too much time taking pictures at Little League games.)

Plays do more than simply allow the operator to change many more specific settings at once, because plays can contain information about the desired end product; the goal for the shot. This enables, the automation to perform intent based actions based on real-time information about the goal and environment where the shot is taken. For example, the automation could be set to take the picture when the ball hits the bat, or the bee lifts off the flower. These events happen too quickly to be reliably captured manually, but can be captured by automation if it "knows" what to look for.

Once the automation knows the goal, it can also determine whether the goal is being met. For example, if there is not enough light, the shutter speeds necessary to stop motion may result in underexposure. Similarly, it may not be possible to expose a scene in a way that captures both shadow and highlight detail. In these cases the camera might warn the operator, suggesting the use of a flash or asking whether it should sacrifice the shadows or the highlights.

2.2 Navigating by Car

Today nearly everyone drives with a navigation system that would have been unimaginable just ten years ago. Enter your destination, and these systems plot a route for you that, to the degree possible, avoids traffic delays. They then provide you with turn-by-turn directions and offer to reroute you as the traffic situation changes.

These systems already provide some HAT-like features. They allow some specification of high-level goals (e.g., mode of transportation, fastest time, shortest route, avoid tolls, avoid highways) and the navigation system will generate route options with estimated driving times. On a computer, you can even create a route manually, and Google Maps, will give you an estimate of how long it will take to drive it. However, in current systems, the list of options is relatively limited. To some extent, this maybe unavoidable. Today's automation may not have a good sense of what makes a road scenic or fun to drive. Thus, if finding a scenic route is a goal, greater input from the operator may be required. However, in choosing a scenic route, the operator presumably would appreciate feedback from the automation about things it does know, for example, time to destination and road closures.

Current navigation systems also fail to give operators crucial information related to what is often the primary question on their minds: Am I going to make it there on time? They *do* give you an estimate of your arrival time based on current driving conditions, and often a fairly useless reason for any delay (e.g., "Traffic is heavy"). However, driving in traffic is generally not so simple. Apple, Google, and Waze have access to large databases from which they could generate statistical profiles that would allow them to answer questions like: How early do I have to leave in order to have a 95% chance of being on time? What is the probability that I hit traffic on the Bayshore Freeway if I leave at 3:00? What alternatives do I have if traffic gets worse?

The answers to these questions would allow a user to develop a plan for a trip that goes beyond the routing currently provided, to develop alternatives in case problems develop in transit. Current navigation systems offer to reroute you if traffic patterns change and another route becomes faster. Unfortunately, while you are driving is not the time for "bi-directional communication" with your navigation system; carefully vetting the proffered route is difficult (and possibly illegal) while driving. On the other hand, simply accepting this offer, can be a bit of a crapshoot. You might end up zipping along a highway, but you might find yourself in a warren of little streets in a questionable neighborhood. A solution to this problem might be to move the bi-directional communication to before departure. If you live in a large city with traffic problems, you probably find yourself periodically discussing your commute with co-workers. You probably discuss your strategies for getting home. Leave by 4:00, take one freeway unless it is unusually slow, in which case switch to surface streets. Maybe a co-worker has suggested a new better route. We suggest that navigation systems could become like very knowledgeable co-workers (at least knowledgeable when it comes to traffic) sitting in the right seat looking at the bigger picture. Using the navigation system, you could develop a strategy for your commute. This strategy does not have to be static. Maybe you have a preferred route and are only willing to change if you can save ten minutes. Once en route, maybe you are willing to be rerouted from one freeway to another to save five minutes, but only want to switch to surface streets if it will save ten minutes. Maybe you only want to switch if there is a 90% chance that the new route will actually end up being faster. Maybe you want the automation to ask before switching you to surface streets but simply to reroute you when it finds a faster highway. These strategies look a lot like the plays discussed above. They can be very complicated. However, because they can be formulated offline, they allow you to direct how to adapt to changing driving conditions without requiring you to negotiate with the navigation system while you are driving. Further, because these plays can be reused every time you drive to and from work, investing some time in developing a good one can save significant amounts of time down the road.

3 HAT Design Patterns

Across aviation, photography, and automobile navigation, we see very capable automation that does not achieve its full potential because it is not aware of the goals and expertise of its human operators. In each case, default parameters are set by designers and engineers that assume a set of generic goals on the part of the user. These defaults are difficult or impossible for the operators to modify. It is often unclear what they even are. While the human could, in many cases, ignore the automation, this would sacrifice important abilities the automation has that could improve outcomes. In each case we propose an interface that allows the operator better access to modify these parameters by specifying goals in a more nuanced way, by providing transparency into how the automation will meet those goals, and by allowing for negotiation with the automation when those goals cannot be met. Here we attempt to capture what generalizes across these domains.

3.1 Plays

One solution that appears to be useful across all the domains we have looked at is plays. Plays encapsulate goals, procedures, and division of responsibility into a package that can be specified offline and instantiated quickly in real-time situations. Plays help to realize our tenet that the operator is in control by allowing the operator to explicitly request a course of action quickly, reducing the need for automation to guess at the operator's intent. Plays do this by shifting much of the communication about context and authority (see Structure in Sect. 3.5 below), offline. We see this in the development of the play for photographing baseball where team colors and field position are entered before the game, and in the navigation example where various route options and their priority are entered before departure. Plays can also help with transparency, for instance, in the automobile navigation case, the plays make the priorities used by the automation explicit.

3.2 Timing

One interesting generalization between these examples is the effect of timing. In both the photography and the navigation example, there is a planning phase, where specification of the set of relevant plays occurs, and an execution phase. The execution phase itself consists of discrete action events (taking of pictures; path changes), with pauses between them. Changes to the play can occur between these actions, but would be disruptive during execution.

3.3 Bi-directional Communication

Another solution that seems to generalize across domains is bi-directional communication. Much work has gone into the proper allocation of functions between automation and human operators (e.g., [4, 7]). However, in human-human teams, team-members often perform similar if not identical functions; just imagine a brainstorming session. More formally, with traditional "Pilot Flying/Pilot Monitoring" procedures in aviation, a second person is used to generate ideas and catch errors more than to add new functionality. Interestingly, there is a similar style of programming, called pair programming, where two programmers sit together at one monitor, one typing code and the other monitoring for errors. We see something similar in both the photography and navigation discussions above. Both the automation and the human operator may have a role in performing a particular function, potentially the same role. For example, in photography, the operator can focus at roughly the correct distance and let the automation fine tune, but sometimes the automation may focus and the operator may need to fine tune. Similarly in navigation, the operator and automation may go back and forth fine tuning a cross country trip to go along scenic routes and visit particular locations while also meeting a timetable and reaching a camp ground each evening. While the human sets the mission level goals, even at that level automation may have input as to whether the goals are achievable. We see this sort of back and forth, bi-directional communication, as a critical part of making human-computer interaction into teaming. Thus, we believe the development of interfaces that support bi-directional communication is crucial for HAT.

3.4 What Is a Design Pattern?

We have been discussing two HAT-inspired solutions to common problems with automation: plays and bi-directional communication. In other fields, such generalized solutions to common problems are often captured as "design patterns." Design patterns were introduced in architecture by Alexander, et al. in the influential book A Pattern Language: Towns, Buildings, Construction [11]. For example, the pattern Raised Walk, is offered as a solution to the problem "Where fast moving cars and pedestrians meet in cities, the cars overwhelm the pedestrians. The car is king, and people are made to feel small." Design patterns have been particularly influential in computer programming. The book Design Patterns: Elements of Reusable Object-Oriented Software [12], introduces 23 patterns, following a more elaborate format than A Pattern Language. Each pattern is broken down into sections specifying (among other things) the intent, motivation, when to use it, consequences, related patterns, and advice on implementation.

In conjunction with the NATO working group on Human Autonomy Teaming (HFM-247), we have been working to develop similar design patterns for HAT. These patterns are evolving as members of the working group attempt to define them in ways that will be useful to their current projects, support generalization to new projects, and interact well with each other. Below we give a preliminary sketch of a bi-directional communication pattern, based on the observations above. This sketch follows an abbreviated version of the format used by Gamma et al. [12].

3.5 A Bi-directional Communication Pattern

Intent. First, our pattern lays out a brief description of what the pattern attempts to do. Our bi-directional communication pattern supports generation of input from all relevant parties and its integration into decisions.

Motivation. Next comes a description of the problem and how the pattern solves it. From the above examples, it is apparent that for many problems humans and automation bring differing strengths and weaknesses to a problem. Looking at the task of focusing the camera from the photography example, automation is generally faster and more accurate than human operators. But automation can focus on the wrong object or fail to find an object to focus on entirely. The operator can supply information that improves the autofocus's performance and add information when the autofocus none-the-less gets it wrong. The result is a system that is less error prone than either operator or automation by itself. The situation with navigation is very different, yet, in many ways, very similar. Again the automation has important strengths. It can pull together great stores of information and make them available to the user. However, because navigation systems work in a domain filled with uncertainties (often behaving in a complex and non-linear way), they cannot provide certain answers. Above we imagine a future system that provides a more detailed statistical description of the various options available to the operator. Still, it falls on the user to decide what types of risk to take.

Further, the examples given here show the advantages of making communication bi-directional with information going back and forth between the parties (as opposed to a simpler system in which, for example, the automation simply alerts the operator to some information).

Applicability. The examples cited above show the potential utility of implementing the bi-directional communication pattern in cameras and navigation systems. Our studies in aviation have shown that a back and forth between humans and automation results in solutions that are more acceptable to the human operators [2, 13]. Bi-directional communication facilitates sharing of information regarding problem definition, potential solutions and authority to act, information that has been shown to be critical in a wide variety of situations [5, 8, 14–16]. Conversely, it is important that automation be designed to take advantage of human knowledge and expertise. Automation can only react appropriately within the range of situations it was developed for. Outside this range, it may lack access to relevant information or the ability to generate appropriate plans, making it brittle. Allowing human operators to input information improves the system's flexibility. Thus, we believe this pattern is broadly applicable to complex automation.

There are some exceptions, however. Bi-directional communication may not work with all types of automation such as genetic algorithms and neural networks, because these systems lack the structure necessary to provide a rationale for their ratings and recommendations. Also, there are situations where it may be necessary to limit communications. This is particularly true in urgent situations where time is not available for comprehensive communications.

Structure. Next we describe, abstractly, what the solution looks like. For our bi-directional communication pattern, we are concerned primarily with what types of information need to be shared between automation and the human operator. We have divided the information that needs to be shared between automation and the human operator into three: Authority (what level of automation should the automation be working at?), Context (what problem/what goals are the human and automation attempting to work on?), and Options (how could we achieve our goal or solve our problem?)

Authority. One crucial piece of information that needs to be communicated between automation and the human operator regards authority. This could be as simple as whether the operator or automation is performing certain tasks. At a slightly more complex level, we can imagine assigning the automation intermediary levels of automation (LOA) [17]. For example, the automation might propose a course of action (e.g., the navigation system proposing a re-route) which must be accepted by the operator. As automation gets more complex, however, we envision a more complex authority structure. For instance, automation may be assigned more complex "working agreements" like contingent LOAs (e.g., reroute me to surface streets if such a reroute is predicted to save more than 10 min). Using such automation, human operators will need access to the current working agreement and will need the ability to dynamically change task allocations and levels of automation.

Context. Before developing a plan, automation and human operators must communicate about the context in which the plan is being created:

- What is the goal? Ordinarily we would expect the operator to set the goal (Where do I want to go?), but even here the automation may play a role (e.g., if a system failure is detected on an aircraft, the automation might propose diverting [2]).
- What are the constraints? A given situation typically comes with some constraints that rule out certain solutions (or, at least, make them categorically worse than others). For example, an aircraft cannot divert to an airport that is outside its fuel range, and a car cannot drive the wrong way down a one-way street. Both the operator and the automation may be aware of constraints on how the goal might be achieved and need to be able to convey this information to the other party. There may also be temporal constraints, time limits by which actions must be taken.
- What are the priorities? In addition to constraints, other factors can be more naturally traded-off against one another. If a passenger has a heart attack, you want to transport him to a good medical facility as quickly as possible. That is great if the closest airport also has the best medical facilities, but if it does not, how much time are you willing to give up in order to get better facilities? Weights allow you to define a function across the different factors that go into your decision, communicating to the automation how much the operator cares about these different factors.

Options. Of course the reason you have the automation is to calculate options. This could be at a very low implementation level (e.g., auto-throttles of an aircraft adding thrust to maintain airspeed), or at a mission level (e.g., a route planning tool suggesting where to divert to). The automation may be authorized to implement options without operator input (aside from the initial authorization). However, the operator should have access to these options, the rationale for selecting them, the projected consequences of their implementation, and the automation's confidence in these outcomes. The operator should also be able to generate options and have the automation evaluate those as well.

Implementation. What guidelines are available for implementing this pattern? Bi-directional communication takes time. If one had to negotiate with the autofocus system before capturing an action shot, or a route-planning tool while driving in traffic, it would not happen. The examples given above suggest, however, that some or all of the communication can be done offline. Plays offer a means of encapsulating and abbreviating this communication allowing the operators to specify when and how decisions are to be made before the urgency of real-time operations sets in. Even when the play is in progress, there are points of greater and lesser urgency. Interaction should be scheduled between shots in the photography example or turns in the case of navigation. In implementing this pattern for other domains, designers should be conscious of similar rhythms.

4 Next Steps: Toward a Framework

HAT has been a goal since the dawn of the computer age [4]. However, today, with self-driving cars on our streets and self-flying aircraft in our skies, we have a much clearer picture of where automation is heading. While things that were only dreams a short while ago have quickly become indispensible, our interactions with the automation are often frustrating. Human factors engineers, unfortunately, are playing catch-up in trying to shape a more satisfying relationship with these automated systems. In this paper we present a snapshot of our strategy for developing a framework for HAT. Our strategy begins with tenets derived from CRM. CRM aims to maintain clear lines of command and authority while fostering free and open exchange of relevant information. We started from a position that the human operator should remain in control, and that goals, plans, and information relevant to accomplishing those goals and plans should be freely shared. We then asked, what does the operator remaining in control and sharing this information look like in practice? Our goal is to iterate this process, updating our tenets based on our exploration of their implications for the design of real systems. In doing so, our goal is to develop a framework for HAT, consisting of our tenets, guidelines for implementing the tenets, and software libraries that make this implementation easier. To achieve these goals we must be able to find generalizations in how human operators effectively use automation. We see the development of HAT interfaces as being parallel to the development of graphical user interfaces in the 1980s or touch interfaces earlier this century. In both cases there was an early period of experimentation, which eventually settled into a familiar set of design elements (e.g., desktop, windows, and menus). Software frameworks developed that mirrored these design elements allowing for easy reuse, and accelerating adoption. The kind of intelligent automation for which HAT would be useful is still in its infancy.

We expect fluidity in HAT interface design, until the underlying automation matures. However, we expect the kind of interaction discussed here to become increasingly prevalent in the years to come.

Acknowledgments. We would like to acknowledge NASA's Safe and Autonomous System Operations Project, which funded this research. Thanks to Walter Johnson, Vernol Battiste, Mike Matessa, and Gilles Coppin, for their thoughts and comments.

References

1. Shively, R.J., Lachter, J., Brandt, S.L., Matessa, M., Battiste, V., Johnson, W.: Why human-autonomy teaming? In: Proceedings of the 8th International Conference on Applied Human Factors and Ergonomics (2017, this edition)
2. Brandt, S.L., Russell, R., Lachter, J., Shively, R.J.,: A Human-autonomy teaming approach for a flight-following task. In: Proceedings of the 8th International Conference on Applied Human Factors and Ergonomics (2017, this edition)
3. Strybel, T., Keeler, J., Mattoon, N., Alvarez, A., Barakezyan, V., Barraza, E., Park, J., Vu, K.-P., Battiste, V.: Measuring the effectiveness of human automation teaming in reduced crew operations. In: Proceedings of the 8th International Conference on Applied Human Factors and Ergonomics (2017, this edition)
4. Licklider, J.C.R.: Man-computer symbiosis. IRE Trans. Hum. Factors Electron. **1**, 4–11 (1960)
5. Endsley, M.R.: From here to autonomy: lessons learned from human-automation research. Hum. Factors **59**, 5–27 (2017)
6. Chen, J.Y.C., Barnes, M.J.: Human-agent teaming for multi-robot control: a literature review. Army Research Lab Technical report, ARL-TR-6328 (2013)
7. Feigh, K.M., Pritchett, A.R.: Requirements for effective function allocation: a critical review. J. Cogn. Eng. Decis. Mak. **8**, 23–32 (2014)
8. Lyons, J.B.: Being transparent about transparency: a model for human robot interaction. In: AIAA Spring Symposium Series (2013)
9. Woods, D.D.: The risks of autonomy: Doyle's catch. J. Cog. Eng. Decis. Mak. **10**, 131–133 (2016)
10. Miller, C.A., Parasuraman, R.: Designing for flexible interaction between humans and automation: delegation interfaces for supervisory control. Hum. Factors **49**, 57–75 (2007)
11. Alexander, C., Ishikawa, S., Silverstein, M., Ramió, J.R., Jacobson, M., Fiksdahl-King, I., Angel, S.: A Pattern Language: Towns Buildings Construction. Oxford University Press, New York (1977)
12. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Boston (1994)
13. Battiste, V., Johnson, W.W., Dao, Q., Brandt, S.L, Johnson, N.H., Granada, S.: Assessment of flight crew acceptance of automated resolution suggestions and manual resolution tools. In: 26th International Congress of the Aeronautical Sciences, Anchorage, AK (2008)
14. Lee, J.D., See, K.A.: Trust in automation: designing for appropriate reliance. Hum. Factors **46**, 50–80 (2004)
15. Hoff, K.A., Bashir, M.: Trust in automation: integrating empirical evidence on factors that influence trust. Hum. Factors **5**, 407–434 (2015)

16. Wiener, E.L., Kanki, B.G., Helmreich, R.L. (eds.): Crew Resource Management. Academic Press, Cambridge (2010)
17. Parasuraman, R., Sheridan, T.B., Wickens, C.D.: A model for types and levels of human interaction with automation. IEEE Trans. Syst. Man Cybern. Part A: Syst. Hum. **30**, 286–297 (2000)