

POLICY AS AN INTERACTION METHOD FOR DECISION SUPPORT SYSTEMS

Michael C. Dorneich, Stephen D. Whitlow, Christopher A. Miller, John A. Allen
Honeywell Laboratories
3660 Technology Drive, Minneapolis, MN 55418

This paper introduces the notion of *policy* as a basis for interaction for decision-support systems, and describes how policy was applied as the foundation of a decision-support tool to aid in diversion management in airline operations. A *policy* is an abstract, general, a priori statement of expressing a goal and an associated priority. Diversion management is the process of deciding which incoming flights to divert and to which airport they will be diverted. The consequences of diversions can be complex due to the interdependent nature of resources and schedules, and the multiple stakeholders that are impacted by the decision. In current practice there is little consideration of how diversion decision will impact airline operations due to the difficulty of acquiring and analyzing the relevant data. Policy is used to capture the goals and priorities of all stakeholders and when used as the basis of a critiquing decision-support system, thereby ensures inclusion of their interests into the decision making process.

INTRODUCTION

A *policy* is an abstract, general, a priori statement expressing a value or goal and some notion of the priority of that goal. In its simplest form, policy provides a method for human operators to mathematically define what constitutes "goodness". A set of individual policy statements can be bundled together, and these *policy bundles* can be used to flexibly define the priorities that apply in a given situation (priorities can change given different circumstances). One domain that represents a highly constrained system where the specific situation greatly impacts the optimal strategy is airline flight and dispatch operations, and thus decision-support systems in this domain would benefit greatly from a flexible definition of priorities that is context-dependent.

Airline dispatchers are responsible for routing flights and keeping them on schedule. However, the schedule is frequently subjected to disruptions due to unpredictable factors like weather, mechanical failures, and other unforeseen circumstances that affect the airline's ability to execute the schedule as planned. If a flight is unable to land at its original destination, dispatchers must decide to which airport to divert that flight. Diversion decisions have dramatic consequences in terms of disruption to the airline's four interlinked schedules: (1) aircraft fleet schedule, (2) crew schedule, since crews often move from one aircraft to another in the course of their working day, (3) maintenance schedule, with aircraft arriving at maintenance bases for scheduled maintenance checks, and (4) passenger schedule. There are other stakeholders in the diversion decisions as well, such as marketing (don't divert a flight that has been heavily marketed as a new reliable service), or station operations (do not overload one airport with too many diverted flights). Accordingly, multiple stakeholders are impacted by the diversion decision made by the single decision-maker, the dispatcher. The dispatcher's decision is driven by safety concerns only. Fuel limits are, obviously, the most important criterion in determining diversion decisions, since aircraft

must be safely on the ground well before their fuel expires. This is currently the most important, and in many cases, the only criterion upon which diversion decisions are based. There are typically, however, multiple diversion plan possibilities, all that will maintain safe flight and landing profiles, and yet which differ widely in their impact on various aspects of airline operations, profits, and customer convenience and satisfaction.

Thus within the space of safe operations there are better and worse decision that can dramatically affect the airline's ability to recover from the disruption and get back on schedule ("recover the schedule"). Current practice is characterized by limited access to information in a timely manner, due to information being spread across different systems and different departments. Due to time pressures and their stated priorities, dispatchers do not consistently take into account how the decisions will impact airline operations. Additionally, the "quality" of diversion decisions often is dependent on the experience level of the dispatcher, where more experienced dispatchers have learned about the priorities and operations of other sectors of the airline and take these into consideration when making diversion decisions. What is needed is a way of capturing the goals and priorities of the other stakeholders, and using this information to inform the dispatcher's decision.

This paper will describe the Diversion Off-Gate Management Assistant (DOGMA) system, a critiquing tool that uses policy as the basis upon which to provide decision support to help airlines make diversion decisions that, while retaining safe operating practices, will also go further toward providing decision makers with the broad and diverse set of concerns from various stakeholders in their diversion decisions. DOGMA integrates multiple information sources to improve dispatchers situation awareness of the current state of flight, aircraft, maintenance, crew, passenger schedules. Policy can capture the goals and priorities of all interested parties in the diversion decision, thereby integrating their interests into the decision making process. This broader awareness of the various concerns in the decision is, currently, something that is only learned gradually over time. Thus, one impact of

DOGMA is expected to be superior decisions from less experienced diversion managers. Another impact of this increased awareness and broader input into the diversion decision should be better and more consistent diversion decisions, which translates into minimizing the impact of time-critical diversion decisions, and an increased ability of airlines to recover from severe schedule disruptions.

SOURCES OF INFORMATION

Policy

In our sense, a 'policy' is simply a means for a human expert to assert statements, in a mathematically useful format, about what should be sought or avoided in the domain. Individual policy statements can be collected into a 'policy bundle' which, collectively, comprises a picture of what counts as 'good' or 'bad' in the domain. Such statements must necessarily be general, but they are a very common form of communicating about the domain. For example, dispatch domain experts will declare that it is generally a bad thing to divert passengers, and the more passengers diverted, the worse it is. On the other hand, it is especially bad to divert international passengers (since their flights are generally less frequent and, in some cases, the airline may be liable for their delays). These are policy statements, and when we can operationally define their terms and assign a weighting to their value, they become useful in a policy-based approach such as we outline. For example, we might chose to operationalize the policy statements above as follows

- For every passenger arriving at their destination gate more than 15 minutes later than scheduled, count 2 'badness' points.
- For every international passenger arriving at their destination more than 15 minutes later than scheduled, count 2 additional 'badness' points.

Policy statements may be thought of as rules in a production system, but their conclusions must only be arrayed along a mathematical scale. They are heuristics, but they are heuristics with an asserted, quantitative value. As such, it may be more useful to regard them as a decomposed elements of an objective statement or weighted sum—or even as pre-compiled database queries with an associated scoring metric. Each policy statement asserts that a certain condition is good or bad to a certain degree. Later, when a bundle of policy statements is evaluated against a given situation or scenario, the combined goodness or badness of the situation can be ascertained *given the policy statements which have been previously made*. To date, we have only experimented with monotonically increasing or decreasing sums, but there is little reason, in principle, why this must be the case. Our emphasis has been on creating a vocabulary that makes policy statements easy to make and use. Because we have, to date, used policy primarily to inform critiquing and visualization systems, there is little need to ensure that the set of rules is complete or covering, or that it provides a unique recommendation with a defined stopping criterion. Instead, policy statements are bundled into sets of concerns articulated

by perhaps multiple users. These sets of concerns can then be applied, collectively or individually, to a situation or scenario—primarily to see which ones, and how many of them, trigger. In this sense, our use of policy statements is very distinct from the traditional use of rules in a rule-based expert system.

What counts as good or bad can change from situation to situation, company to company, season to season, etc. While safety is always the most important goal in airline operations, there may well be times and situations where some secondary goals, such as passenger on-time arrival, are less important than other secondary goals, such as crew duty limits, and vice versa. This fact is all too often ignored in the creation of decision aiding and resource optimization systems, leaving the operator to either slavishly obey the limited set of considerations the aid reasons over, or to go through extensive mental work to interpret the aid's recommendations in light of what really 'counts' today. It is important to be able to flexibly redefine and re-apply the definition of 'good'. By separating out the definition of "goodness" and collecting it in a policy bundle, the various tools that use it (e.g. visualizations, resource optimization algorithms, schedulers, etc.), may be flexibly redefined to reflect whatever counts as good (or bad) in the current context to the current operator. Policy bundles can be used in a context-dependent manner, where the particular situation may vary the type of policy used. For example, one policy statement we heard articulated in user interviews during DOGMA design was as follows: "it is always better to minimize delays for first class customers, unless you are starting a new weekend service (called a "protected market") such that getting coach class passengers to their destinations on time may be more important than first class ones". Context-sensitive policy statements could be collected and packaged together in bundles to have them organized around real-world situations or could be changed continuously on a rule by rule basis. For example, normal operations vs. pre-holiday operations would be associated with different policy bundles. For instance, it is much more important for passengers to reach their destination (even if late) during the holidays, while during normal operations it is more important for the majority of passengers to be on time (even at the cost of some passengers not reaching their destinations). Figure 1 illustrates two policy bundles for airline operations, one representing the 'badness' of various circumstances under normal operations and one representing the badness of those same conditions on holidays. Since different bundles of policy statements can be created a priori, different policy bundles can be asserted at the time the decision is to be made (i.e. runtime). In other words, which policy bundle is used to evaluate a situation can depend on the situation itself, on the user's preference or on what the user wants or needs to learn about the situation. Users can choose to examine the goodness of the current situation, or of a hypothetical future situation, under 'maintenance's policy' vs. 'marketing's policy', or under the perspective of what counts as good for a holiday vs. a normal operating day. Secondly, since users may, but do not have to, define policies at runtime, there is little extra workload to configure the automated support tool to behave correctly given the current context.

| Policy | Normal Operations Bundle | Holiday Bundle |
|---|--------------------------|----------------|
| Do not exceed crew duty limits | 10 | 8 |
| Do not divert international connecting passengers | 8 | 7 |
| Do not delay flights greater than 15 minutes | 8 | 3 |
| Do not cause passengers to fail to reach destination (even if late) | 3 | 8 |
| Do not divert a flight with an unaccompanied minor | 10 | 10 |
| Do not divert to an airport that has its maximum capacity of aircraft | 8 | 8 |
| Do not divert a flight in a protected market | 4 | 1 |

Taken together, the fact that multiple policies can be created ahead of time and applied to a visualization or decision support tool means that policy provides the tool with an adaptable interface (Opperman 1994). Adaptable interfaces are good because they allow a more precise fit of the support provided to that needed in context, and because they keep the control of adaptation under the human's authority. On the other hand, adaptable interfaces have the problem of increasing user workload (cf. Miller, 2000). With policy, however, adaptations are pre-compiled, thereby shifting the added workload to a less constrained time period. The optimization parameters are not defined by the designer of the tool, but rather by the policy bundle chosen at runtime, hence users are allowed to "finish the design" (Vicente, 1999; Rasmussen, Pejtersen, & Goodstein 1994).

To date, we have experimented with the use of policy both to inform an automated resource allocation system (prioritizing the use of military communication resources—Funk and Miller, 2000) and, as described here, to inform a visualization and decision support system. Of the applications, the latter places fewer constraints on the organization of policy statements in bundles. When automated decisions are to be made using policy, criteria of completeness and consistency must be maintained over the set of policy statements and/or unifying principles must be created and applied when multiple policy statements, perhaps from different users, conflict. Khaneman, Slovic and Tversky, (1982) among many others, have questioned human ability to maintain such constraints in mathematical formalisms without extensive effort and checking. When used as a critiquing system (Smith, McCoy and Layton, 1997), however, policy statements are simply triggered and reported, either singly or collectively, when they are encountered in a situation. As such, this use of policy fulfills more of a reminding role than a true decision making role, and the need for formal organization of policy statements is reduced.

Interaction Design

After deciding to apply policy to the domain of airline operations (specifically diversion management), the next issue was determining how dispatchers should interact with a policy-based system. We considered having the system automatically generate diversion solutions that would subsequently be reviewed by dispatchers; however, research

has indicated that traditional decision support systems suffer from brittleness (the system model does not account for all possible scenarios) and complacency (over-reliance on system recommendations biases operator to not consider some factors and accept computer recommendation without adequate review) (Smith, McCoy, & Layton, 1997; Parasuraman, Molloy, & Singh, 1993). Moreover, domain issues such as FAA regulations, the dispatchers' union, and organizational personality precluded this option as well. We also considered another common problem: that users of automated systems are reluctant to relinquish control to the automation (Miller & Goldman, 1997).

Considering these aforementioned issues we decided to leverage a policy system via a critiquing approach in which dispatchers' input to a diversion plan would be reviewed by a computer partner that would offer feedback if there was problem with the input. This system could be characterized as an interactive critiquing system that has been shown experimentally to be an effective form of decision support (Guerlain et al, 1999). Other research has indicated that interactive critiquing systems have less obtrusive interaction styles than traditional decision support which results in greater user acceptance (Langlotz & Shortliffe, 1983).

Finally we considered which stages of the decision process to automate. Parasuraman, Sheridan, and Wickens (2000) proposed that automation can be applied to four broad classes of functions that correspond to stages of human information processing: Information Acquisition, Information Analysis, Decision and Action Selection, and Action Implementation--acquisition, analysis, decision, and action. . Acquisition is very time consuming in airline operations since most of the data is managed by awkward interfaces to legacy systems (Avient Technical Report, 2001). Dispatchers frequently do not even have time to access the relevant information. The automation in our tool assumes responsibility for acquiring, integrating, and presenting information to the dispatcher. The efficient information presentation acts as an "information manager" that maintains dispatchers' situation awareness (M. Lewis, 1998) Furthermore, by automating acquisition and analysis of specific policy feedback you reduce the amount of information --thus minimizing dispatcher workload for integrating information. Automated policy-based analysis also provides a projection capability to assess the impact of current diversion decision (e.g. diverting flight 318 to SCO would cause the crew to exceed their daily duty limits); this capability has been shown to improve operator situation awareness in a process control environment (N. Moray (1997)

By automating acquisition (presentation) and analysis (policy feedback) we are in-line with the research that suggests cognitive processes that are involved in complex information analysis, such as inferencing and attention, can be negatively affected by stress that is ubiquitous in airline operations (Wickens & Holland, 2000). By making the dispatcher an active decision maker we minimize the potential for negative impacts on situation awareness seen in systems in which the human operator passively critiques computer generated solutions (Kaber et al, 1999). The following is an example of the human-machine interaction based on four-stage human information processing model:

- Automated **acquisition** enables presentation of relevant information to allow human operator to develop situation awareness during early stage of diversion situation.
- Dispatcher does preliminary **analysis** then begins building diversion plan.
- Automation further **analyzes** intermediate dispatcher decisions, such as diverting flight 318 to SCO
- Automation provides policy feedback if dispatcher decision violates policies, dispatcher **decides** whether to modify plan based on consequences feedback from policy.
- Automation also integrates by assigning a score to each candidate plan that enables dispatchers to compare across multiple plans.
- Dispatcher reviews plan(s), **decides** which one to put into **action**

Automating the information acquisition and analysis strikes a nice balance between reducing workload, improving situation awareness while ultimately leaving the dispatcher in complete control of the system.

FINDINGS

Design Process

The research prototype DOGMA was developed utilizing a human-centered design approach. Requirements were gathered through extensive interviews with the target user group (dispatchers) as well as various stakeholders to inform design requirements. Interview transcript statements were put into an indexed table, categorized into one of the following: actions, policy statements, strategy, info requirements, responsibilities, motivations, operational process, organizational culture, design suggestions, and other. A secondary category would be listed when appropriate. A sub-category, or subject, was listed and could be one of the following: metric, pre-planning, scope, consequences, resources, collaboration, definitions, priorities, causes, recovery, regulatory, integration requirements, deployment, training. The table could then be sorted on category or subject and was used to develop use cases. All use case elements were referenced back to the table to ensure that traceability to original user comments. A modified, iterative Rational Design approach (Jacobson, Boock, and Rumbaugh, 1999) was employed to plan the development process. After initial use case development, paper prototypes were developed. Paper prototypes consisted of initial sketches of design ideas for both graphical user interface and the interaction design. A second round of interviews with stakeholders presented the paper prototypes to elicit feedback, comments, and suggestions. The sketches were deliberately kept rough and unpolished to emphasize the extreme malleability and plasticity of the design state. The hope was that the users would not "hold back" their comments in any sense due to the assumption that the system had already been largely designed. The approach helps to pull stakeholders into the design process, gives users something tangible to react to when introducing entirely new decision-support tool

concepts, and makes the user community a part of the design team without letting individual idiosyncrasies distract from the task focus. Interview data was added to the categorized tables, and use cases were developed in more detail, prioritized for development, interactions were identified, and were related to the architectural design. An iterative development plan consisted of which use cases to design, implement and test at each iteration.

Initial interviews with airline operations personnel identified the need for decision support tools to support the situation awareness of dispatchers. What was needed was a common data view across the airline (e.g. central operations, station operations), where presently data access is difficult and inconsistent across functional areas of operations. On the problem of diversion management, one airline supervisor related that "it's [diversion management] always been a problem. [In] 26 years, my entire career has been in dispatch, and there's never been a good way of managing [it]." One typical example of the problems of situation awareness and lack of collaboration is coordination of alternates by dispatchers, where during a major disruption too many planes are diverted to a station. One major airline's dispatch director summed up the needs as follows, "The biggest thing we can give the dispatchers is information: ...[so] they can make their choice based on better information rather than just where's the flight coming from and where's it going." In addition to better information, dispatchers need tools at the operation end that enable the airline to recover from multiple diversions. A dispatcher said, "We know we're going to get hit, don't know when or where, but what you need is the tools to recover when you do." The key point made by interviewees over and over is the ability to see the associated affects of diversion decisions. We received near unanimous endorsement for the application of policy to the diversion management domain as a way to quickly understand the effects of decisions on downline operations and the ability of the airline to recover. The domain experts resonated to the idea of having stakeholders' voices present at the decision point. They felt that "the more people involved in a decision the better both economically and for safety."

DOGMA Prototype

The DOGMA prototype is divided into two principle spaces: (1) the Information Space, and (2) the Diversion Plan Workspace. The information space, found on the left half of the interface, provides an integrated view into the information available (e.g. flights, crews, maintenance) and can present multiple views of the information (Map Display, Schedule View, Tabular View). The primary goal of the information space is to maintain the dispatcher's situation awareness across multiple information sources by allowing for rapid access to relevant information. The right side of the interface contains the Diversion plan workspace where dispatchers construct diversion plans by selecting aircraft and deciding where they should be diverted. Dispatchers are presented with a set of policy violations relevant to the diversion plan. The associated penalties are added up to give a total score for the plan. By viewing the policies, dispatchers are informed of the

airline priorities that the plan violates, and dispatchers can choose to modify the plan accordingly. For instance, a decision to divert flight 123 may violate the policy of "Do not divert a flight with an unaccompanied minor on board". The dispatcher did not need to know an unaccompanied minor was on board until that fact impacted his or her diversion decisions. In this way policy is used to present relevant information to a dispatcher only when he or she needs it.

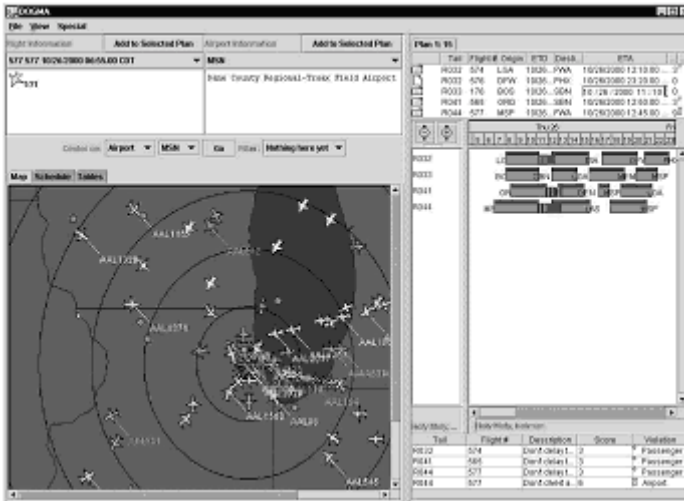


Figure 2. Diversion Off-Gate Management Assistant

DISCUSSION

This policy-based critiquing system addresses many of the domain's usability challenges. For instance, the system enables asynchronous collaboration and data integration by presenting the voices of multiple stakeholders to the single decision-maker at the moment of decision. Moreover, critiquing reduces workload by providing feedback about only those policies that are violated by a user action. The system further reduces workload by enabling a quick comparison of diversion options by providing a simple metric - total policy penalty points on a selected flight. This approach also accommodates both novice and expert dispatchers. The interaction does not interfere with an expert's workflow unless a decision violates a policy; and policy feedback provides learning experiences for novice dispatchers. Consistent outcomes are also insured by the systematic and automated evaluation of user actions.

The use of policy has enabled the efficient and seamless integration of enterprise-wide goals at the decision point, thus enabling organizational control and effect over decisions. Furthermore, policy systems facilitate the propagation of high-level enterprise goals, such as customer satisfaction, down to the operational level via relevant policy feedback (e.g. don't delay a passenger twice on a trip). Finally, such a system improves visibility into other stakeholders' priorities thus minimizing "bunker mentality" within departments of an organization.

ACKNOWLEDGMENTS

The authors would like to acknowledge Jim Richardson, Jim White, Jon Schewe, Ranjana Deshpande, Rahima Kaba, and Trent Reusser for their contributions to this project.

REFERENCES

Avient Technical Report (January 2001) : Avient Assessment and Recommendations Report. Internal Honeywell Technical Report.

Funk, H., Miller, C., Richardson, J., Johnson, C., and Shackleton, J. (2000). "Applying Intent-sensitive policy to automated resource allocation: command, communication and most importantly, control". In *Proceedings of the 5th International Conference on Human Interaction with Complex Systems*. April 30-May 2; Urbana, Ill. 179-183.

Guerlain, Stephanie; Smith, Phillip J.; Obradovich, Jodi Heinz; Rudmann, Sally; Strohm, Patricia; Svirbely, John; Sachs, Larry. "Interactive critiquing as a form of decision support: an empirical evaluation". *Human Factors*, Vol. 41(1), March 1999, pp. 72-89.

Jacobson, Ivar, Boock, Grady, and Rumbaugh, James (1999). *The Unified Software Development Process*. Addison-Wesley, Reading MA.

Kaber, D.B., E. Omal, and M.R. Endsley (1999), "Level of automation effects on telerobot performance and human operator situation awareness and subjective workload," in *Automated Technology and Human Performance: Current Research and Trends*. Mahwah, NJ: Erlbaum, pp. 165-170

Kahneman, D., Slovic, P., and Tversky, A. (1982). *Judgment under uncertainty: Heuristics and biases*. Cambridge Univ. Press; Cambridge, UK.

Langlotz, C.P. & Shortliffe, E.H. (1983). Adapting a consultation system to critique user plans. *Int'l Journal of Man-Machine Studies*, 19, 479-496

Lewis, M. (1998). Designing for human-agent interaction. *Artificial Intelligence Magazine*, vol. Summer, pp. 67-78.)

Miller, C. (2000). The human factor in complexity. In T. Samad and J. Weyrauch, (Eds.), *Automation, Control and Complexity: An integrated approach*. John Wiley & Sons; Chichester.

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

Moray, N. (1997). "Human Factors in Process Control," in *Handbook of Human Factors and Ergonomics*, G. Salvendy (ed). NY: Wiley pp 1944-1971.

Opperman, R. (1994). *Adaptive User Support*. Lawrence Erlbaum, NJ.

Parasuraman, Raja; Sheridan, Thomas B.; Wickens, Christopher D. "A model for types and levels of human interaction with automation." *IEEE Transaction on Systems, Man, and Cybernetics--Part A: Systems and Humans*, Vol. 30, No. 3, May 2000.

Parasuraman, Raja; Molloy, R., and Singh, I.L. (1993). "Performance consequences of automation-induced 'complacency.'" *International Journal of Aviation Psychology*, 3, 1-23.

Rasmussen, J., Pejtersen, A. and Goodstein, L. (1994). *Cognitive Systems Engineering*. Wiley: New York.

Smith, P.J.; McCoy, E.C.; Layton, C. "Brittleness in the design of cooperative problem-solving systems: the effects of user performance". *IEEE Transaction on Systems, Man, and Cybernetics--Part A*, Vol. 27, No. 3, May 1997.

Vicente, K. (1999). *Cognitive Work Analysis*. Lawrence Erlbaum, NJ.

Wickens, C. D. & Hollands, J. (2000). *Engineering psychology and human performance* (3rd ed.). New York: Prentice Hall.