Intelligent Pairing Assistant for Air Operation Centers

Jeremy Ludwig
Stottler Henke Associates
San Mateo, CA
Ludwig@stottlerhenke.com

Eric Geiselman
Air Force Research Laboratory
Dayton, OH
Eric.Geiselman@wpafb.af.mil

ABSTRACT

Within an Air Operations Center (AOC), planners make crucial decisions to create the air plan for any given day. They are expected to complete the plan in part by pairing targeting or collection tasks with the available platforms. Any assistance these planners can acquire to help create the plan in a timely manner would make the entire process more efficient and effective. This paper describes the Intelligent Pairing Assistant (IPA) prototype, which would provide pairing recommendations at specific decision points in the planning process. IPA is designed as a plug-in for software systems already in use within AOCs. The primary contribution described in this paper is the application of existing research in intelligent user interfaces to a novel domain.

Author Keywords

Air operations center, intelligent user interface, reinforcement learning, pairing.

ACM Classification Keywords

H.4.1. Office Automation: Workflow management.

INTRODUCTION

The role of an Air Operation Center (AOC) in the United States Air Force is to provide command and control of air operations. Simply put, the AOC receives a high-level description of tasks and effects and generates a plan of how to best execute them. Within an AOC, planners make crucial decisions to create the overall air plan for any given day. They are expected to complete the plan in a limited amount of time, in part by pairing collection or targeting tasks with the available platforms and weapons. Any assistance these planners, especially the less experienced ones, can obtain to help create the air plan in a timely manner would make the entire process more effective.

One major challenge is in orchestrating the target list; designing packages to strike more than one target at a time makes efficient use of resources and improves survivability. Making the best use of available resources presents another challenge. For instance, rather than assigning a unmanned aerial vehicle to a collection task, it may be more expedient to further task a manned aircraft that is already operating in the area. Hurried human planners often overlook opportunities to take advantage of relationships between tasks.

The idea behind the Intelligent Pairing Assistant (IPA) is to help AOC planners make more efficient and effective use of resources. One existing system allows planners to use a "wizard" system to walk through the planning process. This system helps focus the user on particular parts of the user interface for different planning decisions. However, there is still a wide array of possibilities within each wizard page. The prototype framework illustrates how data already contained in the existing planning system would be used to make *recommendations* to planners in the context of a particular wizard page. That is, IPA provides an intelligent user interface that highlights preferred decisions and certain types of optimizations in order to assist planners in making efficient and effective decisions. Given the potential consequences of these decisions, attempting to foster trust in the recommendations is an important part of the interface. The prototype framework also illustrates how planners can enter *annotations* to specific items on the target and collection lists. This allows planners to associate information not already available electronically with a task, which can be used to help refine the recommendations supplied by the pairing assistant.

The remainder of the introduction is devoted to describing related work. In the next section we identify requirements for IPA. Following this are descriptions of the interface and implementation of the IPA prototype. Next is a section on the annotation interface. The conclusion summarizes this paper and outlines directions for future work.

Related Work

The work described in this paper relies heavily on existing research. This includes representing expert knowledge [e.g. 4], interactive/adaptive recommendation systems [1, 3, 6, 10], and especially research in the area of building trust in recommendation systems [2, 5, 7, 8, 9, 11, 12]. The primary contribution described in this paper is the application of prior research in these areas to a novel domain.

RECOMMENDATION REQUIREMENTS

As stated previously, one of the challenges facing IPA is to build trust within the AOC planning community. Several of the system-level requirements stem directly from attempts to address this challenge. First, the model used to make recommendations needs to be in a human-readable format and small enough that a human could understand it [8].

IUI'12, February 14–17, 2012, Lisbon, Portugal.
Copyright 2012 ACM 978-1-4503-1048-2/12/02...$10.00.

This requirement was used as a guideline throughout the prototype with an emphasis on maintaining simplicity. Second, an explanation facility needs to be incorporated into the user interface so planners understand why recommendations were made [2, 5, 7, 8, 9, 11, 12]. Third, the interface should be responsive to the planner and adjust recommendations based on implicit and explicit feedback [10]. Fourth, the user should have the chance to verify any decisions made as a result of recommendations [7].

The design of the user interface should also take into account existing guidelines for recommendation systems [6]. These guidelines include limiting the number of recommendations, which ensures that they are relevant to the decision being made, and that at least some of the recommendations are "good". Based on these guidelines, we developed two additional requirements for the recommendation table: fifth, the recommendations need to be limited in number, and sixth, recommendations with little utility should not be shown at all.

USER INTERFACE FOR RECOMMENDATIONS

The IPA prototype illustrates how IPA would function as a plug-in to an existing wizard-based planning toolkit. The wizard walks users through the planning process via a number of customized screens that expose only the relevant portion of the tool to the planner as they go about fulfilling planning requests. Generally, IPA is an unobtrusive toolbar that resides on the bottom of the screen. When recommendations are available on the current wizard page, the user can bring up a panel that displays the recommendation interface.

The recommendation interface is based on a common table that displays recommendations created by IPA in a similar format across recommendation types for different decision points. An example table is shown in Figure 1. The beginning and ending columns are always the same, with the intervening columns specific to the particular decision being made. The remainder of this section includes details on the recommendation table and on providing explanations and making user of user feedback.

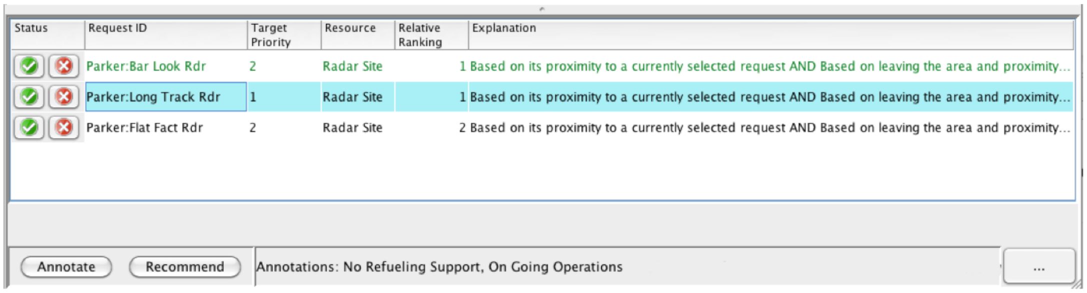
The first column contains the accept/reject buttons. If accept was clicked, the existing AOC software would be expected to be updated with the actions associated with the selected recommendation, the text color turn green, and the buttons become grayed out, as shown in the first row. At this point, the planner could look at the proposed decisions in the existing AOC to verify the result. If reject were clicked, the recommendation would be removed from the table and the user given a chance to explain the rejection by selecting one of a number of available options from a list. Accepting/rejecting will adjust future recommendations even if no explicit feedback is given, by adjusting the relative ranking given to the recommendations. Explicit rejection will be used to manually improve the model

The second to last column is the relative ranking of the recommendations. This is presented to the user as a value of 1 (best), 2, or 3 (worst) in order to maintain similarity to existing priority ranking within AOCs. Note that while relative rank is always seen as 1, 2, or 3, recommendations are sorted based on the actual rank. Therefore, a recommendation with rank of 1.25 would appear before one with 1.33, even though they both appear to have a rank of "1" to the planner. Recommendations with a rank > 3 (significant negative feedback) would not be displayed in the table at all, and those with a rank < 1 (positive feedback) would be shown as having a 1.

The last column is the explanation for the recommendation. This is an English description of the rule or rules that fired to create the recommendation. The point of the explanation column is to allow the user to quickly scan the rationale behind each of the recommendations. The explanation column also includes popups that allow the user to quickly display the complete explanation when it is too large for the table area. Currently, the form of popup interaction is click to show, click to release, though other options would be investigated as part of future work. Ideally, most of the time, the planner should be able to see all of the recommendations and all of the explanations simultaneously. Based on the length of actual explanations in a deployed system, and the expected number of recommendations, we may adjust row height, column width, etc. to minimize the situations in which popups are needed.

DEVELOPING RECOMMENDATIONS

This section describes our investigation into the technology responsible for creating and ordering recommendations. Two particular areas were investigated. The first is knowledge representation – how to represent the state of the world and subject matter expertise in making pairing



Status	Request ID	Target Priority	Resource	Relative Ranking	Explanation
✓ ✗	Parker:Bar Look Rdr	2	Radar Site	1	Based on its proximity to a currently selected request AND Based on leaving the area and proximity...
✓ ✗	Parker:Long Track Rdr	1	Radar Site	1	Based on its proximity to a currently selected request AND Based on leaving the area and proximity...
✓ ✗	Parker:Flat Fact Rdr	2	Radar Site	2	Based on its proximity to a currently selected request AND Based on leaving the area and proximity...

Annotations: No Refueling Support, On Going Operations

Figure 1. IPA recommendation table. The columns headers in this recommendation example are: Status, Request ID, Target Priority, Resource, Relative Ranking, and Explanation.

decisions. The second is developing the feedback algorithm, which empowers individual planners by allowing them to alter the shape of future recommendations.

Knowledge Representation

The reasoning engine is the system that is responsible for creating recommendations by applying a model of expert knowledge to a data model that contains all of the required details of the current planning problem (collection list, resources, weather, etc.). For this project, we decided to use the representation of production (IF-THEN) rules to meet the project requirements:

- Sufficiently powerful to represent decision policies, while still allowing for efficient inference.
- Can be used to provide clear explanations to the user about the rationale behind decision-making.
- Can be verified by subject matter experts prior to deployment. This is especially true in the case of IPA, where the problem is divided into a number of non-interacting rule-bases for specific decision points, and each individual rule-base will contain a relatively small number of rules.
- Can be created manually (working with subject matter experts) or be learned (from compiled knowledge).

As part of the prototype research effort, we modeled a small portion of the rules developed by subject matter experts using the commercial off-the-shelf Jess rule-based reasoning engine (http://www.jessrules.com). The main aspects of the prototype are: a simple data model created in Java, rules created in Jess, running the Jess engine to generate recommendations, and finally, gathering recommendations and supplying them to the user interface. Each recommendation also includes a simple explanation that is constructed by the rules that create the recommendation.

Influencing Recommendations based on User Feedback

In the user interface, the planner will have three options with a provided recommendation: ignore, accept, and reject. Each of these options provides feedback that will be used to adjust the ranking of recommendations, which will in turn affect the ordering and visibility of recommendations in the user interface. Additionally, one of the requirements is that the planner needs to see that their feedback is being used quickly and correctly, in order to help build trust in the recommendations made by IPA. The rank-updating algorithm is designed to respond quickly to feedback, with a goal that the recommendations from a rule should stop being displayed after only three rejections (three strikes and it is out).

Ranking of recommendations is performed by associating weights with the rules that are used to create the recommendations. If only one rule creates a recommendation, then its relative rank is equal to the weight of the rule. In cases where more than one rule is responsible for a recommendation, the recommendations relative rank is determined by averaging the weights of the associated rules.

The approach we took to learning how to adjust the weights associated with rules based on the user's feedback falls into the category of reinforcement learning problems [13]. The proposed algorithm for rule weight update is V(a) = V(a) + r, where V(*a*) is the value associated with a particular rule, *a,* and *r* is the positive or negative feedback given by the user based on the reward function. The initial V(a) is 1, the minimum -3 (performing well), and the maximum 5 (performing poorly) This algorithm is designed to be easy to understand relative to the 1-2-3 ranking already used in AOCs, maintain responsiveness to user feedback and support the desired three strikes functionality.

The amount and direction of the reward *r* is:

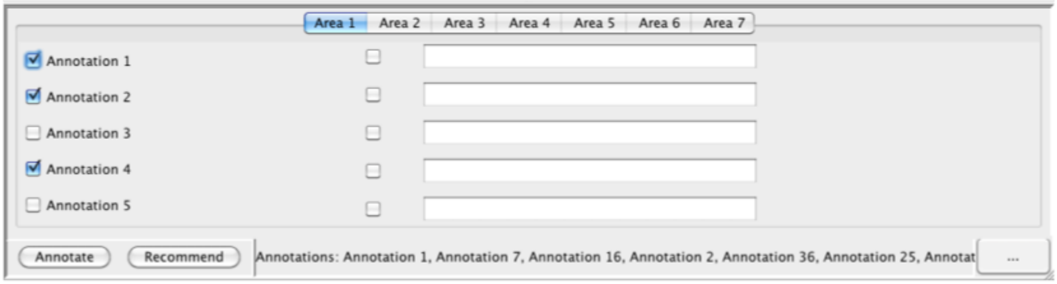
1. (Accept recommendation.) Set *r* ← -1
2. (Reject recommendation without detailed feedback.) Set *r* ← +1
3. (Reject recommendation with detailed feedback.) Set *r* ← +0.75
4. (Implied rejection.) Set *r* ← +0.25

Accepting a recommendation improves the ranking of associated rules (lower is better) while rejection for no reason decrements the ranking by the same amount. If a recommendation is rejected for a particular reason, the associated rules are decremented by a lesser amount, with the idea that they may be generally useful even if they were rejected in this particular context. Finally, an implied rejection occurs when a recommendation is selected that has a worse ranking than another recommendation that was not selected.

ANNOTATION INTERFACE

A second objective of the prototype was to identify how additional information associated with targeting and collection tasks, the "why" behind planning decisions, could be captured. This is information that is not currently available in a computer-friendly format but is required to make good decisions. That is, IPA needs to allow the planner to quickly encode information they might have received verbally, from online chat discussion, or from Word / PowerPoint documents so that the pairing assistant can provide the best possible recommendations.

Figure 2 shows the annotation interface implemented in the prototype, where the actual annotation names and categories have been replaced with placeholders. The interface is designed to be quick and easy to use, with a small number of annotations (less than 50) divided across a set of tabs with familiar names that form distinct categories. In most cases, the planner needs only to check the appropriate boxes to indicate the annotation. Based on the results of our knowledge engineering efforts, this relatively simple interface should be able to cover a majority of the needs usually associated with a task. Additionally, the prototype supports a flexible system that allows planners to



☑ Annotation 1 □
☑ Annotation 2 □
□ Annotation 3 □
☑ Annotation 4 □
□ Annotation 5 □

Annotations: Annotation 1, Annotation 7, Annotation 16, Annotation 2, Annotation 36, Annotation 25, Annotat

Figure 2. Annotation interface. The annotation categories are browsed via tabs (top) while the annotations themselves are selected via checkboxes. A summary of the selected annotations is shown along the bottom, even when the annotation pane is not visible.

suggest new annotations by typing in new annotation names. These new annotations would be recorded and could be used in future recommendation rules.

CONCLUSION

To summarize, this paper presents a prototype of an Intelligent Pairing Assistant designed to allow planners in Air Operations Centers to complete their jobs more efficiently and effectively. We describe the requirements that were developed, the underlying technologies and algorithms, and the assistive user interface.

As this work is in a preliminary state, there is significant future work starting to get underway on nearly every aspect of the system; for example: performing additional knowledge engineering, adding more flexible explanation capabilities, and meeting the technical challenge of integrating the developed system with the actual AOC system. Significant research questions and technical challenges that were not addressed in the prototype exist in each of these areas.

ACKNOWLEDGMENTS

We would like to thank Todd Cloutier, Drew Decker Donald Hulten, Bart Presnell, Beverly Sanford, and Milt Waddell. This material is based upon work supported by the United States Air Force Research Laboratory under Contract No. FA8650-10-M-6082. The views, opinions, and/or findings contained in this article/presentation are those of the author/presenter and should not be interpreted as representing the official views or policies, either expressed or implied, of the AFRL. DISTRIBUTION STATEMENT A: Approved for Public Release, Distribution Unlimited

REFERENCES

1. Adomavicius, G. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering 17,* 6, 734–749.
2. Arnold, V., Clark, N., Collier, P.A., Leech, S.A., & Sutton, S.G. The Differential Use and Effect of Knowledge-Based System Explanations in Novice and Expert Judgment Decisions. *MIS Quarterly, 30*, 1 (March 2006), 79-97.
3. Berry, P., Donneau-Golencer, T., Duong, K., Gervasio, M., Peintner, B., & Yorke-Smith, N. Evaluating user-adaptive systems: Lessons from experiences with a personalized meeting scheduling assistant, in *Proceedings of IAAI'09* (Pasadena CA, July 2009), AAI Press.
4. Buchanan, B. and Shortcliffe, E. *Rule-Based Expert Systems: The Mycin Experiments of the Stanford Heuristic Programming Project*. Addison Wesley Publishing Company, Reading MA, 1984.
5. Ehrlich, K., Kirk, S.E., Patterson, J., Rasmussen, J.C., Ross, S.I., & Gruen, D.M. Taking advice from intelligent systems: the double-edged sword of explanations. In *Proceedings of IUI '11* (Palo Alto CA, Feb 2011), ACM Press, 125-134.
6. Gervasio, M.T., Moffitt, M.D., Pollak, M.E., Taylor, J.M., & Uribe, T.E. Active preference learning for personalized calendar scheduling assistance, in *Proceedings of IUI '05* (San Diego CA, Jan 2005), ACM Press.
7. Glass, A., McGuinness, D.L., & Wolverton, M. Toward establishing trust in adaptive agents. In *Proceedings of IUI '08* (Gran Canaria, Canary Islands, Spain, Jan 2008), ACM Press, 227-236.
8. Fleischmann, K.R. and Wallace, W.A. A covenant with transparency: opening the black box of models. *Commun. ACM 48*, 5 (May 2005), 93-97.
9. Gregor, S. and Benbasat, I. Explanations From Intelligent Systems: Theoretical Foundations and Implications for Practice. *MIS Quarterly 23*, 4 (December 1999), 497_530.
10. Langley, P. User modeling in adaptive interfaces. In *Proceedings of UM '99* (Banff, Canada, June 1999), Springer Wien, 357–370.
11. Lim, B.Y., Dey, A.K., & Avrahami, D., Why and why not explanations improve the intelligibility of context-aware intelligent systems. In *Proceedings of CHI '09* (Boston MA, April 2009), ACM Press, 2119-2128.
12. Pu, P. and Chen, L. Trust building with explanation interfaces. In *Proceedings of IUI '06* (Sydney, Australia, Jan-Feb 2006), ACM Press, 93-100.
13. Sutton, R. and Barto, A, *Reinforcement Learning: An Introduction*. MIT Press, Cambridge MA, 1998.