Knowledge-Based Systems 12 (1999) 443–456

Knowledge-Based SYSTEMS

www.elsevier.com/locate/knosys

The Rotorcraft Pilot's Associate: design and evaluation of an intelligent user interface for cockpit information management

C.A. Miller[a,*], M.D. Hannen[b]

[a]Honeywell Technology Center, 3660 Technology Dr, Minneapolis, MN 55418, USA
[b]The Boeing Company, 5000 E. McDowell Rd., Mesa, AZ 85215, USA

Received 1 June 1999; accepted 18 June 1999

Abstract

The US Army's Rotorcraft Pilot's Associate (RPA) program is developing an advanced, intelligent "associate" system for flight demonstration in a future attack/scout helicopter. A significant RPA component is the intelligent user interface known as the Cockpit Information Manager (CIM). This paper describes the high-level architecture of the CIM, with emphasis on its pilot-perceptible behaviors: Crew Intent Estimation, Page Selection, Symbol Selection/Declutter, Intelligent Window Location, Automated Pan and Zoom, and Task Allocation. We then present the subjective results of recent full mission simulation studies using the CIM to illustrate pilots' attitudes toward these behaviors and their perceived effectiveness. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Cockpit information management; Rotorcraft Pilot's Associate; Associate systems; Page selection; Symbol selection/declutter; Automated task allocation; Pan and zoom; Window location; Intent estimation; Intelligent User Interfaces

1. The Rotorcraft Pilot's Associate program

The US Air Force's Pilot's Associate programs were among the first efforts to implement large, adaptive, intelligent user interfaces (IUIs) [1]. The US Army's Rotorcraft Pilot's Associate (RPA) program is an ongoing technology demonstration extending this work and bringing it to implementation [2]. In this section, we will briefly describe the approach to IUIs we have developed for the RPA with emphasis on the intelligent information management cockpit behaviors, which result in task-sensitive, dynamically generated cockpit configurations.

The Rotorcraft Pilot's Associate program is a five-year, US$80 million research contract managed by the US Army's Aviation Applied Technology Directorate at Ft. Eustis. It currently represents the US Army's largest research and development commitment and is one of the largest ongoing IUI application efforts in the world. The goal of RPA is to develop and demonstrate in flight an advanced, intelligent "associate" system in a next-generation attack/scout helicopter. Associate systems are collections of intelligent aiding systems that, collectively, exhibit the behavior of a capable human [3,4]. They can: (a) perform roughly the same breadth of activities as a human expert operating in the domain; (b) take initiative when necessary, though they generally follow a human colleague's lead; and (c) integrate over ongoing activities to exhibit robust, coordinated, appropriate behavior.

A critical goal of RPA is to manage the mass of information available in future military operations so that the human pilots of an advanced attack/scout helicopter can easily access all and only those portions that are relevant at any given time. Further, RPA must accomplish this without increasing pilot workload or decreasing situation awareness.

1.1. Nature of the RPA task domain

The nature of the RPA task domain differs in many significant ways from domains more familiar to the IUI community. In some instances this makes the job of IUI design easier in the RPA domain; in others, much more difficult.

While other IUI applications e.g. Refs. [5–8] are characterized by truly vast quantities of information (i.e. the whole Web), highly configurable interface and automation technologies, highly variable and unstructured task needs, and comparatively mild time constraints, RPA differs on all these fronts. The information available to a military helicopter pilot is extensive and growing, but it is constrained by available sensors and pre-formatted datalink communications.

* Corresponding author.
E-mail addresses: cmiller@htc.honeywell.com (C.A. Miller), matt.hannen@boeing.com (M.D. Hannen)

0950-7051/99/$ - see front matter
PII: S0950-7051(99)00035-0

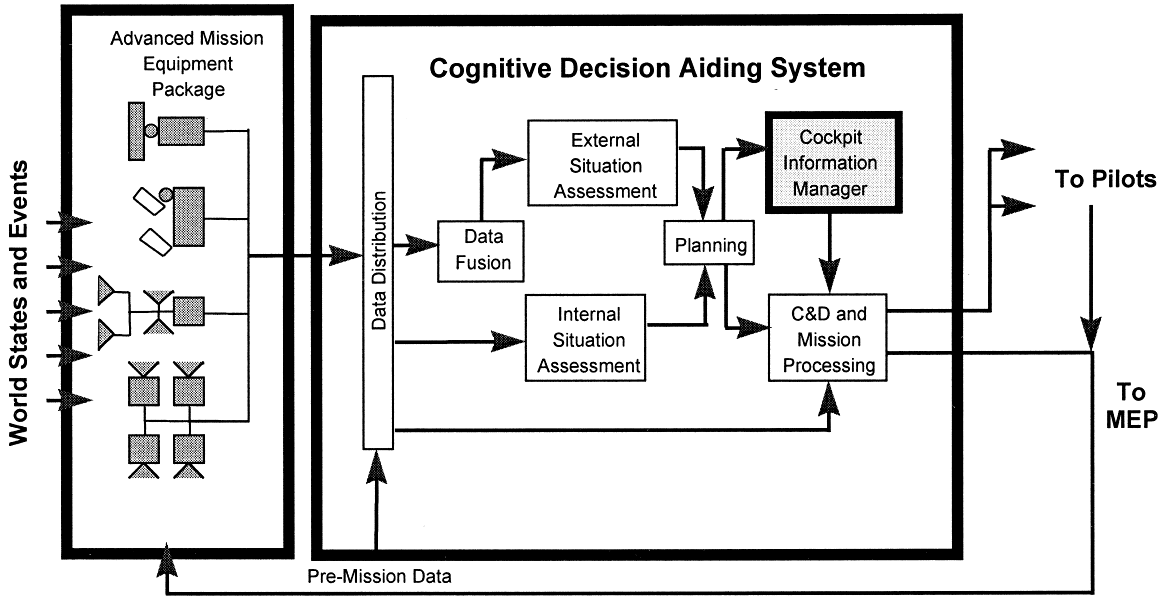

Fig. 1. Functional architecture of the RPA system.

Thus it is not quite as broad and variable than that available to, say, a military operations planner—much less a university student with a Web query. While military glass cockpits are highly flexible, capable of presenting information in multiple modalities and formats, they maintain strict constraints and conventions on information formatting to facilitate transfer of training and ease of uptake. Thus, a truly novel interface, generated dynamically for each situation, is not acceptable, since it would cause too much confusion for a pilot in a potentially critical flight situation. While the Army attack/scout helicopter mission is one of the most flexible in the military, and future battle scenarios call for even more dynamic mission planning and re-planning, procedures are exhaustively thought out at all levels from broad force strategies to specific mission plans and pilots are extensively trained in these procedures. Thus, user tasks are reasonably well scripted and shared, and most situations will have appropriate doctrine created to enable users to ‘fall back on their training’. Finally and importantly, response times are critical. The huge number of potential threats (passive and active) in the operational environment demands constant and complete situation awareness. Flexibility of use, and even operator autonomy, may sometimes be sacrificed if the payoff is faster reaction time and better overall human–system performance.

These aspects of the RPA domain have colored our development of an IUI for it. As is described below, we have emphasized decreasing pilot workload to access and comprehend available information and adhering to the standards and expectations for interface construction, over dynamically accessing novel information or generating novel presentations. Above all, we have emphasized gaining pilot acceptance for the IUI aspects of the RPA cockpit. This paper will report both the IUI design approach we took, and the insights gained from recent full mission simulation evaluations about those aspects of the RPA IUI which pilots are willing and unwilling to accept.

1.2. Overall RPA architecture

Fig. 1 illustrates the whole RPA architecture. There are two major parts to this system. First, the RPA sits on top of an Advanced Mission Equipment Package (AMEP) which provides a suite of sophisticated automated systems including advanced sensors, communications and targeting systems. While highly capable, these are all “traditional” automation systems in that they serve a single function without explicit reference to operator goals and have little autonomous capability. While highly intelligent by the standards of traditional automation, they are what Riley [3] refers to as assistant or slave systems rather than true “associates”. The second major component of the RPA system is the Cognitive Decision Aiding System (CDAS). CDAS integrates the functionality of the Mission Equipment Package with explicit models of crew tasks to sew the traditional automation systems into a semi-autonomous “associate”.

CDAS itself consists of five modules. Data Fusion is responsible for correlating the stream of incoming data from multiple external sources into a single, accurate and unified view. External Situation Assessment consists of battlefield and target assessment subsystems that reason about the significance of external conditions to known mission goals. Internal Situation Assessment performs similar functions on internal aircraft health and status monitoring equipment. A series of six real-time Planners use this assessment data to offer suggestions for maximizing successful accomplishment of known goals. Each Planner is responsible for a single functional area: route planning,

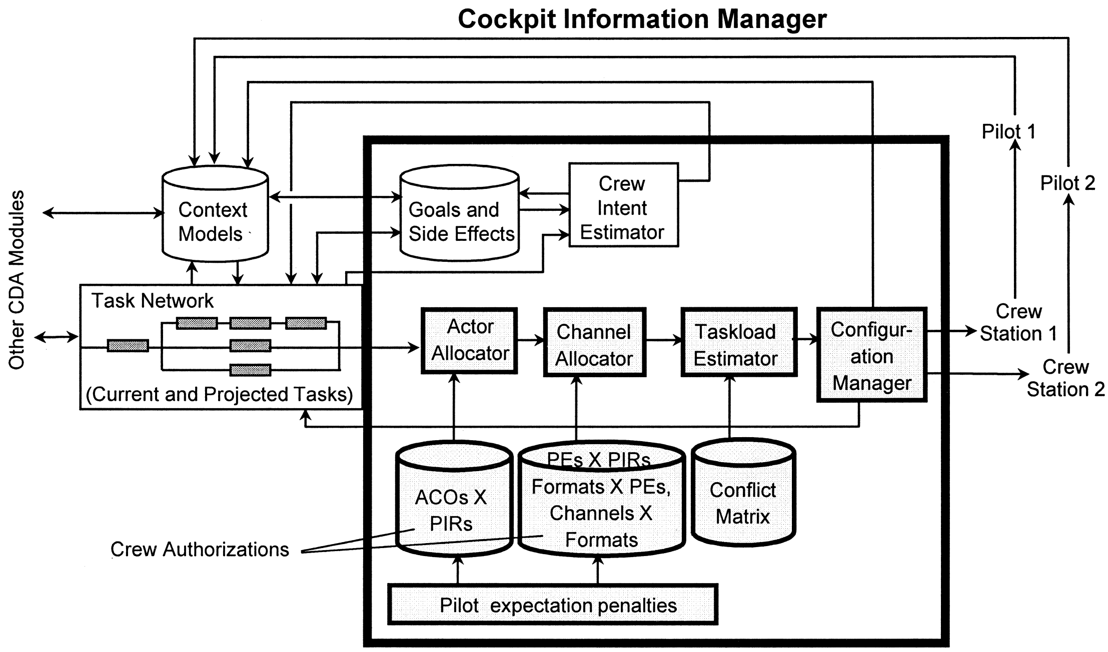

Fig. 2. Functional architecture of the RPA CIM.

survivability, communications, sensor management, attack and reconnaissance. The Cockpit Information Manager (CIM) is the IUI component for the RPA CDAS, thus it will be described in more depth below. First, we will briefly describe the architecture of CIM, then we will present detailed information on the knowledge representation CIM uses and, finally, we will describe the observable behaviors that CIM exhibits and that pilots were asked to evaluate.

2. The RPA cockpit information manager

2.1. Cockpit information manager architecture

The architecture of the Cockpit Information Manager (CIM) is described briefly in this section. For more detailed information, see Refs. [9,10].

CIM is primarily a task-based, rather than state-based, artifact-based, or user-based IUI (cf. Refs. [8,11]). This task-centered focus works well in a highly proceduralized and trained domain such as helicopter operations. CIM is responsible for determining the current and near-future tasks of the crew, and then adjusting the cockpit configuration to meet task needs. A cockpit configuration consists of an allocation of all active tasks to a mixture of cockpit actors (two pilots and automation), and an allocation of the interface functions required for those tasks to some mix of cockpit display and control devices. The ultimate goal of interface management in CIM is enhanced task performance, thus CIM prioritizes and filters presentations so that the most important tasks have their information needs met first, and crew workload and display device capacity thresholds are not exceeded. The Controls and Displays (C&D) and Mission Processing logic is then responsible for issuing the low-level avionics commands to achieve automation tasks and cockpit configurations commanded by CIM.

Fig. 2 illustrates the architecture of the CIM we have developed. Task and context information are provided to the CIM by two shared-memory resources of the CDAS as a whole: the Task Network and the Context Model. The Context Model represents the CDAS' current beliefs about the state of the aircraft and the world. The Task Network represents the CDAS' current beliefs about the tasks that are now being performed and upcoming. In essence, all tasks that can be performed in the RPA aircraft are modeled in the Task Network (cf. Refs. [9,12]), along with alternate methods for performing them. Before takeoff, a mission-specific task network is crafted to represent the mission plan. During a mission, activation and completion conditions on tasks are triggered by pilot actions or world states (recorded in the Context Model), thus enabling the Task Network to maintain, in real time, a model of the active, expected, and completed tasks at any point.

CIM is not just a consumer of task information. The Crew Intent Estimator component (see Fig. 2) interprets pilot actions and world events against mission plans in the Task Network and, using knowledge of goals and side effects, ascertains whether the pilots are following the mission model or are attempting alternate plans or goals. The Crew Intent Estimator can revise the Task Network model to reflect new crew intent to perform a different set of tasks. More details on the Intent Estimator's function and mechanism can be found in Ref. [13].

The set of active tasks, constantly updated and prioritized, is CIM's basis for interface management. CIM's reasoning

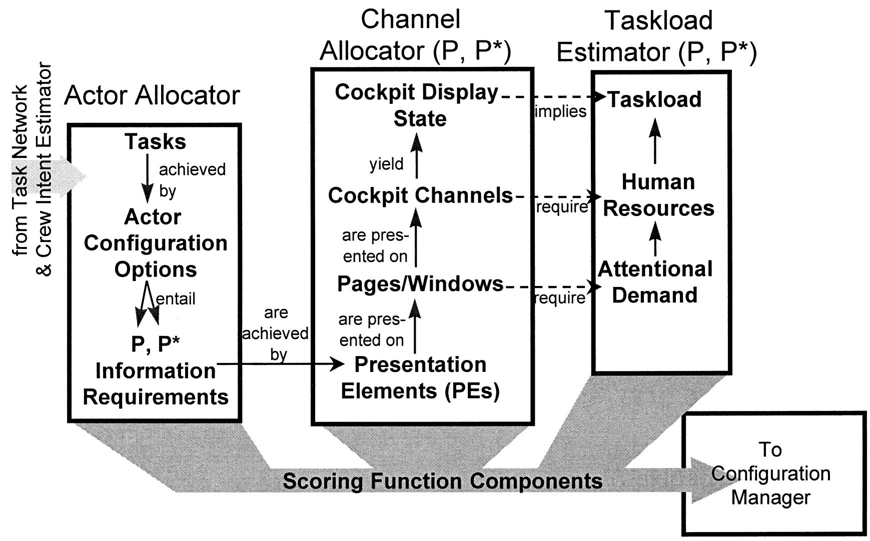

Fig. 3. Knowledge representation and reasoning in CIM.

is conducted in four stages as illustrated in Fig. 2 and in more detail in Fig. 3. The remaining CIM modules are responsible for intelligent management of the user interface. These modules and their roles are described below:

- The *Actor Allocator* is responsible for selecting from among pre-defined legal combinations of actors for each active task. There are three possible task-performing actors in RPA: the pilot (generally on the flight controls and flying the aircraft), the copilot (generally in charge of other mission tasks such as navigation, weapons and communication) and the cockpit automation. Very frequently there will be only one legal combination, and even when there are multiple possibilities, the Actor Allocator may not be permitted (see below) to reassign a task to an actor once the task has been started. For new, not yet initiated tasks, however, and for situations in which significant benefit would be obtained by switching a task from one actor combination to another, Actor Allocator is designed to choose an optimal combination of actors.
- Once an actor allocation has been chosen, the *Channel Allocator* reasons over the set of information that the human actors involved in the task need to have in order to accomplish the task and the set of available presentation methods for conveying that information to the crew members. The details of this complicated process will be presented in more detail in the following Section 2.2. The output of the Channel Allocator is a 'Cockpit Configuration'—an exhaustive stipulation of the configuration of each information presentation device in each of the two RPA cockpits.
- Once CIM has determined who will be doing what and what information presentations they will be using to perform their tasks, it is capable of estimating the taskload effects of those allocations on the human actors. This is the role of the *Taskload Estimator* which uses an algorithm based on Wickens [14] multiple resource theory to estimate the attentional demands imposed by the proposed cockpit task and information configuration—both overall and by individual human attentional channels (vision, audition, psychomotor and cognition). See Ref. [15] for a more detailed description of this algorithm.
- Finally, the *Configuration Manager* reasons over the set of cockpit design decisions which have been made in developing the proposed cockpit configuration and uses the taskload estimation data to provide an evaluation of the overall 'goodness' of the design. This goodness score was initially designed for use in a generate-and-test approach to interface design with the intention that CIM would iterate over proposed cockpit configurations in an attempt to identify ever better designs. In practice, to enhance the speed and performance of the CIM, large portions of the scoring metric have been folded into the search strategies performed by each of the components described previously. This complicated scoring metric will be described in more detail in the following section.

2.2. CIM's knowledge representations and reasoning

Each of the CIM modules described above relies on a large body of knowledge. This knowledge may be divided into four types which, we have argued [9,10] are the kinds of knowledge required for *any* task-based IUI:

- Task knowledge—knowledge about the tasks to be performed, alternate methods of accomplishing those tasks and each task's current status;
- Information Requirements knowledge—knowledge about the information needed to perform each task in each way;

- Source and Presentation knowledge—knowledge about the alternate information sources and methods of presentation available which may be drawn upon to satisfy each information requirement; and
- Evaluation knowledge—knowledge used to determine how good a job has been done in selecting and combining sources and presentations to meet the information requirements of a given set of tasks.

The flow of reasoning in CIM is presented in Fig. 3. The representation and use of each of these types of knowledge will be described in Sections 2.2.1–2.2.4.

2.2.1. Task knowledge

Task information is provided to the CIM by the Task Network. The Crew Intent Estimator module of the CIM has a role in reviewing current Task Network knowledge and updating it based on inferred crew intent. The role and operation of the Task Network and Crew Intent Estimator was described briefly above and further details are beyond the scope of this paper. More details may be found in Refs. [9,12].

CIM refines the tasks described and recorded in the Task Network by selecting among task performance options representing alternate ways the task can be done by combinations of one or both pilots with or without automation. The mechanism for this selection occurs in the Actor Allocator component of the CIM (see Fig. 2). The Actor Allocator draws on a knowledge base of authorized “Actor Configuration Options” (ACOs). Each ACO contains (1) a “Crew Preference Penalty” (ranging from 0 to 3) to be incurred for selecting this ACO and representing the crew’s wishes about doing this task in this way, and (2) a set of parameterized “Pilot Information Requirements” (PIRs) to be met when this task is done in this way. When pilots wish to deny CIM any role in selecting from among alternate actor allocations, they may provide only one legal ACO. When they are willing to allow CIM to reason among alternate allocations, they may tune the degree to which CIM will strive to achieve one alternative over the others by the preference penalties they assign to each option.

2.2.2. Information requirements knowledge

Each ACO provides the Information Requirements knowledge for each actor via its Pilot Information Requirements (PIR) list. We, as Greeno [16] and many others, e.g. Refs. [17,18], construe information as the arguments to a problem solving process represented by the task performance method. These PIRs may be provided in more or less convenient and useful ways, but they must be available if the task is to be performed. PIRs are intended to be abstract representations of information needs, independent of any specific display presentation element.

Simply listing the types of information required for an ACO will generally not be sufficient to enable the selection of a good set of Presentation Elements (PEs) for information presentation. Instead, we need to provide a description of *how* the information is needed for the performance of the task. We provide a set of descriptive “Information Parameters” to meet this need, based on work by Geddes et al. [19,20]. Each PIR includes five parameter/value pairs describing how that information is needed for this method of performing the task:

- Importance—the relative necessity of this information for task success;
- Scope—the proportion of the total range of values for this information which must be presented simultaneously during performance of the task;
- Resolution—the level of detail required for the information during task performance;
- Bandwidth—the frequency with which the data must be sampled during the plan;
- Control—the need to affect the value of the information during task performance (rather than simply monitoring it).

Each parameter is assigned a value from 0 to 10 to indicate greater or lesser requirement of that attribute for the PIR. For example, a precision maneuvering activity (such as a landing) might require altitude information with high resolution and bandwidth but relatively low scope, while a more relaxed maneuver (such as normal flight operations) might require larger scope but lower resolution and bandwidth.

The method for knowing which PEs can meet PIR needs, for assessing how well they meet those needs, and for knowing how they can be combined in the cockpit are described next. A more detailed description of each parameter, and of their use in identifying needs x presentation matches is contained in Ref. [9].

2.2.3. Source and presentation knowledge

Once actors are assigned to tasks and the parameterized PIRs needed for performing those tasks are assessed, the CIM must propose a set of Presentation Elements (PEs) for meeting the information needs. Furthermore, the set of PEs proposed must be organized in combinations (known as ‘formats’ or ‘pages’) which (1) obey good human factors practice, (2) are familiar and usable by the pilots, and (3) are realizable in the cockpit. Finally, the selected groupings of PEs must be placed on selected “channels”—physical devices in the cockpit. These three tasks are performed by the Channel Allocator as depicted in Figs. 2 and 3.

Channel Allocator begins by accessing a knowledge base of Presentation Elements. PEs are the smallest controllable elements which can be manipulated by the Controls and Displays Logic module of the CDAS. For example, if the C&D module can only turn all threat symbols on and off on a map display, but cannot turn individual symbols on and off, then all the threat symbols together represent a PE, even though they may be realized as a set of separate graphical elements.

PEs are represented in the same formalism as PIRs in order to enable selection of presentation methods which satisfy current information needs. Each PE is capable of presenting information which satisfies one or more PIRs to a greater or lesser degree. A PE is defined by the information elements it conveys, and these information element names are the same ones used to characterize PIRs. Thus, a simple lookup can tell CIM which PEs are candidates for meeting a given PIR that is currently active. But each information element conveyed by a given PE is also described in terms of the same information parameter values used to describe the PIR: Scope, Resolution, Bandwidth and Control.[1] Thus, a match computation (described in Ref. [9]) can be performed to determine not just whether, but also *how well* the information provided by the PE meets the needs described in the PIR. In addition to information parameter/value pairs, a Crew Preference penalty score is associated with each PE X PIR X ACO combination to represent the pilot's expectations and wishes about using this PE for this task.

PEs are selected for use in a tentative cockpit configuration on the basis of their fit with the PIRs required by the ACO. Proposed PEs must be organized appropriately, however. One way of accomplishing this is by defining "formats" which are predefined, 'legal' combinations of PEs. Formats are represented in terms of the set of PEs they are capable of displaying and may also contain heuristics constraining the location, configuration and compatibilities of the PEs within them. While an IUI system could be built to exhibit great flexibility in organizing PEs to meet task needs (essentially generating formats "on the fly"—Ref. [17] is an example), this is not always desirable. Formats are a convenient way of ensuring that collections of PEs obey good human factors practice and they preserve expected information organization, thus ensuring some conservatism in cockpit behavior—an important consideration for pilots in highly complex task domains.

The selection of a PE will generally commit the CIM to the selection of a format—which is to say, to a set of constraints to which other PEs can or must be presented along with the one chosen. There may be a one-to-one correspondence between some PEs and their formats (e.g. audio voice messages), but it will be more common for formats to be capable of presenting multiple PEs in various combinations. Formats must be placed on a specific channel, as described below. Crew Preference penalties for the use of a given format for a given task are also recorded at this level.

The final step for the Channel Allocator is to select the channel on which each format is to be presented. Channels are physical I/O devices such as the Left, Right and Center Multi-Function Displays, audio voice synthesizer, 3D audio tone generators, etc. Each channel will be capable of presenting only some of the total set of formats and PEs. A voice PE cannot be presented via a visual display, for example. Again, Crew Preference penalties are recorded for using a given channel to present a specific format.

Information is also recorded about the human attentional resources required to attend to a given PE and the display resources required to present it. Display resources are represented as a percent of total screen channel capacity and are therefore a rough indicator of screen clutter. Human resources, defined in terms of multiple resource theory [14], are vision, audition, speech, psychomotor, and/or cognition. An attentional demand value is recorded for each PE, representing the degree of human attention required to process the information conveyed. Resource utilization and attentional demand values enable the calculation of human workload by the Taskload Estimator (see Fig. 2).

None of the above discussion references where the information provided by a PE is coming from. Although reasoning about information sources is a critical aspect of some domains (e.g. military intelligence and library information retrieval), it is somewhat less important for cockpit operations. We have not, to date, explicitly represented knowledge about information sources separately from knowledge about information requirements and presentation capabilities. In part, we are able to avoid reasoning about these elements due to the overall reliability of information sources in the cockpit, and to their transparency to the pilot. In general, the pilot does not care whether his altitude information comes from a barometric altimeter, a radar altimeter, or some other source. Furthermore, by the time a PE is available for presentation in the cockpit, its source information is deemed reliable enough for flight operations—or its reliability is built into the PE itself and, therefore, becomes a separate information element described for that PE. Nevertheless, this representation and manipulation of information source knowledge is an important avenue for future growth.

2.2.4. Evaluation knowledge—the CIM scoring function

The scoring function used by the Configuration Manager component of the CIM represents a critical aspect of the overall CIM design. Every interface design effort requires myriad tradeoffs, thus the Configuration Manager incorporates an explicit tradeoff philosophy in its scoring function. By representing the goals of good interface design in this domain, and then assessing candidate designs against those goals, we accurately reflect and manage the tradeoffs which an interface designer, whether human or automated, must make.

We interviewed a variety of rotorcraft pilots and RPA designers to develop the consensus list of goals for a "good" cockpit configuration manager which is included in Table 1. We were very much aware that there were incompatibilities between these goals. For example, it is impossible for a dynamically managed cockpit to provide

[1] Importance is not represented for PEs since it makes little sense to describe the 'importance' with which information is presented. Instead, importance is factored into the match computation [20] between PEs and PIRs—effectively weighting the value of getting a good match by the importance of the information need.

Table 1
Relative weighting of consensus goals for a "good" cockpit configuration manager

	Cockpit management principle	Saaty wt.
1	Pilot in charge of tasks	NA
2	All needed tasks accomplished	NA
3	Pilot in charge of info presented	0.235
4	All needed info provided	0.219
5	Stable task allocation	0.143
6	Only needed info provided	0.098
7	Tasks allocated as expected	0.091
8	Info presented as expected	0.070
9	Stable info configuration	0.066
10	Tasks allocated comprehensibly	0.050
11	Only needed tasks active	0.028

all needed information and to be completely stable; it is also practically impossible for all needed tasks to be accomplished and for tasks to be allocated only in a stable or expected fashion.

The first two principles were generally agreed to be inviolable. As such, we built them into the CIM as absolutes: (1) pilots always have the opportunity to access any control function and begin execution of any task they deem appropriate; (2) the CIM system always reasons over all active tasks and satisfies their information needs on the basis of task importance or priority. Furthermore, tasks above a designated critical priority threshold are always serviced, regardless of potential violations other goals.

In order to determine the relative priority and weighting of the remaining goals, we asked the same individuals to weight each factor using the Saaty [21] paired comparison technique. This technique produces not only a rank ordering of factors, but also a relative weighting. The values shown in Table 1 represent an average across respondents.

The weighted configuration management goals presented in Table 1 represent the behaviors for which CIM should strive, as well as an indication of the relative importance of each goal. The evaluation and scoring performed by the Configuration Manager also require, however, a way to link those goals to observable or computable aspects of a proposed cockpit configuration.

This was provided by creating a set of computed parameters representing the degree to which the proposed cockpit configuration violated each of the goal behaviors. We adopted the philosophy of penalizing designs for their failings instead of rewarding them for their accomplishments, though we suspect that either approach would work equally well. The set of parameters and the method of assessing them from the CIM knowledge representation is presented below:

1. *Pilot commanded information violation*—Penalizes changes to Pilot commanded cockpit state based on magnitude of change. Pilot commanded state is recorded as those Channel X Format X PE combinations which the pilot explicitly commanded via the controls. Magnitude of deviation is measured according to a set of defined heuristics (e.g. changing location of a PE within a Format is much less significant than changing Formats). This penalty degrades with elapsed time since the pilot commanded the display state.
2. *All needed information violation*—Penalizes unmet PIRs. Value of penalty increases with the priority of the task with which the PIRs are associated, the importance value of the PIR to the task and degree of PIR/PE mismatch (as measured by incompatibilities in the Scope, Resolution, etc. parameter values).
3. *Stable task violation*—Penalizes any ACO changes over those assumed in the prior configuration. The greater the change (e.g. from one pilot alone to the other pilot alone vs. from one pilot to the same pilot plus automation), the greater the penalty.
4. *Only needed information violation*—Penalizes extraneous PEs (those not required by active tasks) which may be present due to a Format choice or to support stability. Also imposes penalties for excessive information, as measured by channel capacity and human workload predictions.
5. *Expected task allocation violation*—Aggregates Crew Preference penalties for chosen ACOs.
6. *Expected information allocation violation*—Aggregates Crew Preference penalties for all PE, Window, Format and Channel choices.
7. *Stable information violation*—Penalizes any CIM-commanded change in cockpit state over prior configuration.
8. *Comprehensible task allocation violation*—not currently used.
9. *Only needed task violation*—Penalizes task allocation (and, indirectly, information support) for tasks below a low priority threshold.

Each candidate cockpit configuration is scored on each of these parameters, providing an indication of how badly the proposed configuration violates each associated goal. These scores, however, are on widely differing scales (due in part to the historical development of the representation and in part to what was convenient for the various parts of the information representation). In order to be aggregated into a final, overall "goodness" rating, they need to be combinable. This process is illustrated in Fig. 4.

The first step in making the violation penalties

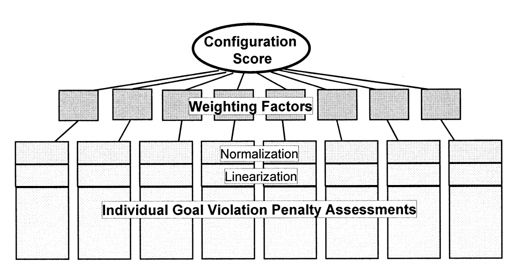

Fig. 4. Approach to aggregating violation penalties.

combinable is to linearize them—that is, to ensure that *within each penalty* each unit represents a equivalent delta in the impact of that scale. For example, although in the information representation the scale for recording Crew Preference Penalties ranges from 0 to 3, the semantics assigned to that scale are such that a '3' (defined as a 'very serious violation of expectations, done only in critical contexts. A pilot might never encounter a situation warranting this severe a violation of expectations outside of combat or serious equipment failure') is much more than three times as serious a violation as a '1' (defined as 'an acceptable but dispreferred alternative. Things might be done this way occasionally for convenience, but it should not be the norm.'). Thus, expectation penalties are linearized by increasing the scalar distance between values to reflect this difference (currently, an assigned penalty of '3' is reassigned to '10').

The second step is normalization, ensuring *across penalty* equivalencies—that is, ensuring that a one step delta on one violation scale is equivalent to a one step delta across all other scales. We are accomplishing this by normalizing at an "average worst case task" level—we define a realistic worst case value for each penalty within a hypothetical worst case task scenario and then compute the ratio of an assessed penalty against this hypothetical worst case value. This should ensure that each assessed penalty value represents a normalized contribution to the overall success or failure in achieving the goals of configuration management.

Having linearized and normalized, we now know the relative degree of violation of each of the Configuration Manager's behavioral goals, but each design goal is not equally important. The Saaty weighting results tell us the relative importance of violating one goal versus another. Thus, the final step in our computation is to take the linearized, normalized value for each penalty and multiply it by its Saaty weight. The sum of linearized, normalized and weighted violation penalties is a single metric for the overall goodness of the proposed cockpit configuration against the set of current crew needs. If time permits, the CIM algorithm can repeat its search, generate new candidate configurations, score them and compare the single metric which results in an attempt to find better configurations.

2.3. CIM interface management behaviors

For communicating CIM's capabilities to pilots, we found it helpful to describe CIM behaviors from an operator's viewpoint instead of, or before, presenting architectural and representational descriptions like those above. CIM performs six primary activities observable by the pilot. These are:

1. Intent Estimation;
2. Task Allocation;
3. Page (or Format) Selection;
4. Symbol Selection/Declutter;
5. Window Placement;
6. Pan and Zoom.

We will describe each of these behaviors below in the context of a representative mission scenario (depicted in Fig. 5). This scenario covers roughly 2 min of mission time. It begins with an "Ingress" task as a part of a larger reconnaissance plan. During this task, the crew encounters an unexpected threat which triggers a previously unplanned (and inactive) "Perform Actions on Contact" task. Subtasks under "Perform Actions on Contact," in this scenario (others are possible) include performing a "Low Probability of Detection Maneuver," a "Sensor Scan" to determine the nature and extent of the threat, and then submitting a "Spot Report". When these tasks are accomplished, the Crew Intent Estimator determines that the crew intends to engage the threats and therefore asserts a newly active "Attack" task, which in turn contains the subtasks "Select Combat Position", "Select Weapons", "Engage Targets" and "Battle Damage Assessment". These tasks are followed, in this scenario, by an "Egress" task.

2.3.1. Intent estimation

The intent estimation behavior of the CIM has little visible output to the RPA crew, but it can dramatically affect the other CIM behaviors through the set of active tasks it reports. At the beginning of our scenario, the "Perform Actions on Contact," task is neither active nor is it a scheduled part of the mission plan. Detection of an unexpected enemy threat (e.g. through Data Fusion's interpretation of sensor data) is an automatic Task Network trigger for the "Perform Actions on Contact," task, but sensors can be wrong and there are battlefield threats that are not detectable with current technology (e.g. small arms fire). Since it is important for CDAS to remain 'in the loop' with the pilot even in these instances, the Crew Intent Estimator constantly tracks crew behavior to infer their intent. Thus, while sensors might be unable to detect a small arms threat, the pilot's pattern of evasive maneuvering, weapon slewing, communications, etc. and will be evaluated using template-based plan recognition techniques [13] to infer the presence of an otherwise undetectable instance of "Perform Actions on Contact".

While intent estimation is mostly invisible, it is not entirely so. The RPA cockpit includes an 'intent interface' to provide the crew with both insight into, and some control

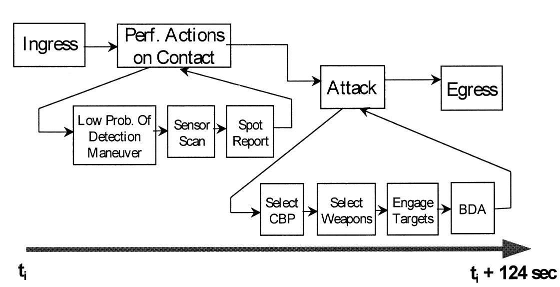

Fig. 5. Representative mission scenario illustrating CIM behaviors.

over, CIM's understanding of their intent. This "Crew Coordination and Task Awareness" display consists of four small LED buttons located in the upper right portion of each pilot's main instrument panel. It reports, in text, the current inferred (1) high-level mission context, (2) highest priority pilot task, (3) highest priority copilot task, and (4) highest priority CDAS task. Pressing these buttons permits the pilot to override CIM's current inferred tasks and assert new ones via push button input. Since the inclusion of a direct method for viewing and interacting with intent estimation was a new development in the RPA cockpit (over prior associate system work), we were especially interested in how pilots would regard it.

2.3.2. Task allocation

Under this behavior, for not-yet-active tasks, CIM determines how best to allocate the task among a 'legal' (i.e. preapproved) mixture of human and automation actors—an ACO. For example, the pilots may have authorized CIM to consider allocating the "spot report" task either (a) to the copilot alone, (b) to automation with the final review and approval being performed by the copilot, or (c) to automation alone with no copilot approval. Each of these would then be a 'legal' configuration option which CIM will consider, though the pilots may also indicate that one method is heavily preferred (a factor included in CIM's evaluation metric and traded off there against potential benefits such as taskload reduction).

2.3.3. Page selection

Given each actors' tasks, CIM determines the best set of pages (i.e. formats) to present on the three available multi-function displays for each crew member. In our scenario, when reacting to an unexpected threat (that is, when the "Perform Actions on Contact," task becomes active) the crew will have high need for information about the threat and about the maneuvering capabilities of the aircraft. Thus, in this context, the three multi-function displays will generally be configured to show a sensor page, a tactical situation map display page and a flight page with primary flight symbology—though the concurrent presence of other, high priority tasks might result in a different configuration. When the decision is made to engage the threat (when the "Perform Actions on Contact," task ends and the "Attack" task begins), weapons configuration and control information are temporarily more important for the copilot, and thus his cockpit may be re-configured to present a weapons page instead of the flight page (see Fig. 6).

2.3.4. Symbol selection/declutter

For each of the selected pages, CIM determines the best set of symbols for meeting the current information needs, removing unnecessary symbology when it taxes the capacity of either the display or of the human to process. For example, during the "Ingress" and "Egress" tasks, the crew needs navigational information and their map displays will

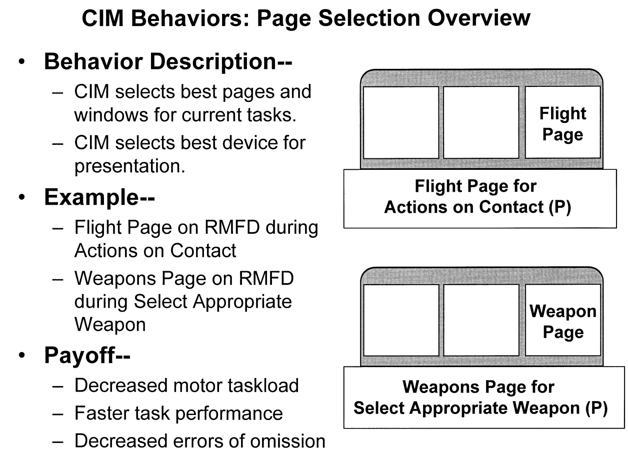

Fig. 6. Page Selection example.

generally show routes, phase lines, passage points, etc. When an unexpected threat appears, however, and the crew begins executing "Perform Actions on Contact," this navigational information becomes less critical than information about the threat and its relative position. Thus, CIM will generally suppress navigation symbology and replace it with threat symbology such as threat icons, lethality and intervisibility envelopes, etc. (see Fig. 7).

2.3.5. Window placement

The RPA crew station can present many types of information in pop-up windows overlaid on a portion of a larger page. Whenever pop-up windows are used, CIM determines their best placement to minimize obscuring other needed information and symbology, yet adhering as closely as possible to expected locations for each window type. For example, when the pilots decide to engage the threats (to begin the "Attack" task), they need to select a combat position. In support of this "Select CBP" task, the AMEP provides them with a recommended position and an explanation of the utility of this position according to the Army's standard position evaluation criteria. This explanation is presented in a pop-up window as nine criteria scores. While useful, placing this large window in any default position on the map risks obscuring threat symbology critical to the ongoing attack task. CIM dynamically selects a position for the window that minimizes obscuration and, when impossible, ensures that only lower importance symbology is obscured (see Fig. 8).

2.3.6. Pan and zoom

CIM also controls the pan and zoom settings of the tactical situation (i.e. map) display to ensure presentation of important symbols, yet avoid clutter. For example, during Ingress and Egress, pilots need a large area presented on the map, though high-resolution for terrain is less important. When a pop-up threat appears, needs shift: high resolution for the area around the threat and possible maneuver paths is important, though the total area shown may be reduced.

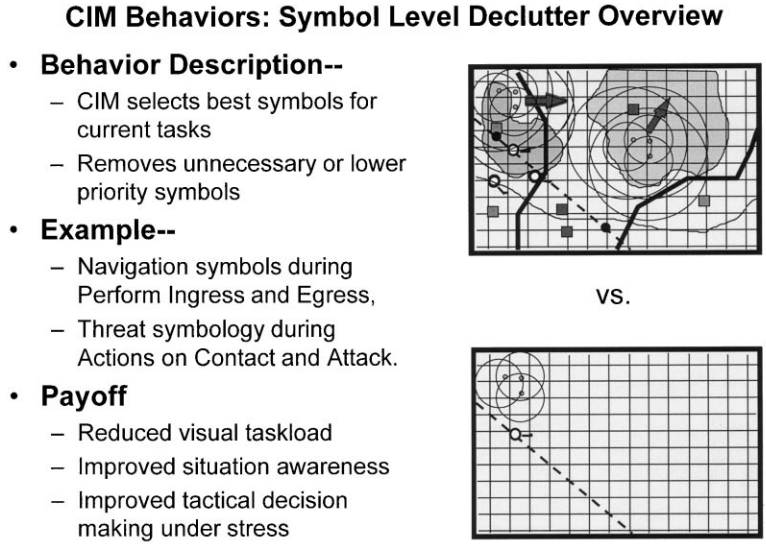

Fig. 7. Symbol Selection example.

CIM reasons through these various needs and asserts pan and zoom commands that adjust the map to appropriate pan and zoom settings.

3. Pilot acceptance of the RPA CIM-simulation test results

The CIM design described in the previous sections has been partially implemented in an RPA Cognitive Decision Aiding System (CDAS) produced by Boeing Helicopters in Mesa, AZ. The Advanced Mission Equipment Package (AMEP), CDAS and RPA cockpit design has been implemented in an Apache helicopter platform and is currently undergoing a flight demonstration phase. Here, however, we will report on data collected during extensive full mission simulations carried out before the flight phase to help evaluate CDAS behaviors and implementation, and to assist in prioritizing issues for flight demonstration. Below, we report the experimental design and primary findings with regards to pilot acceptance of the CIM IUI design. Quantitative performance data are still being analyzed and will be the topic of a future paper.

The RPA Simulator is shown in Fig. 9. The simulator was fixed-base with surround dome visuals. The cockpit and controls and displays were identical to those used in the flight demonstration vehicle and, thus, were of very high fidelity. The simulated test environment included extensive passive and active threats, and human control of the Tactical Operations Center, friendly artillery, and 1–3 wingmen. Realistic communications, including change of mission Fragmentary Operations Orders, were maintained between these players.

The RPA Simulator Evaluation emphasized military mission realism. The 4 Crews (8 pilots) trained and flew together, as they do in field operations. Crews were given realistic objectives and permitted to make their own tactical decisions about how to achieve them. Each pilot received an average of 10.8 h of training in the simulator and 13.9 h of classroom training over a two-week period.

Each crew flew 14 part-mission test scenarios of 20–50 min duration—7 with the RPA CDAS, and 7 with the AMEP alone. The focus of these part-mission scenarios was on exercising a particular CDAS or AMEP behavior and thus context, objectives and task flow were permitted to be somewhat fragmentary or unrealistic. Each crew also flew four 1–1.5 h full-mission scenarios—two with the AMEP alone and two with CDAS. Full-mission scenarios were designed to be highly realistic and crews were given free reign to pursue their commander's objectives via whatever methods they thought appropriate.

The AMEP vs. CDAS conditions were chosen to balance the evaluation over a common baseline of advanced automation technologies. The only difference between the two conditions was the addition of the integrative, associate and IUI technologies of the CDAS (cf. Fig. 1). All missions were

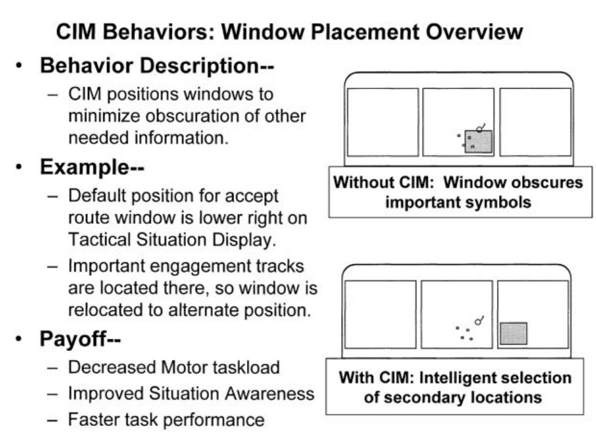

Fig. 8. Window Placement example.

Fig. 9. The RPA Simulation environment.

balanced for complexity. Crews flew the two AMEP or CDAS full missions in sequence and then switched technology conditions and flew the remaining two missions with the other set of technologies. The sequence was counterbalanced to minimize training effects.

The simulation tests were constructed to include numerous examples of the CIM page selection, window location, pan & zoom, and symbol selection behaviors in a variety of tactical mission contexts. Crew Intent Estimation was implemented for the Actions on Contact task alone, and CIM task allocation behaviors were not implemented in the simulation due to time and budget constraints.

To obtain crew acceptance data, a questionnaire was administered to the pilot and copilot after each of the final AMEP and CDAS full-mission test trials. All of the questionnaire responses utilized complete verbal anchoring and a linear response scale with five equal intervals, in accordance with Ref. [22]. The pilots were instructed to mark one point anywhere along the linear scale from 1 to 5. The criteria value (for satisfactory CIM behavior) established before the simulation test was an average score of 3.5.

This criterion was met for three of the four CIM behaviors tested directly. The average and range of pilots' ratings of the behaviors, is presented in Fig. 10. In general, pilots found the CIM behaviors to be 'Of Use' or 'Of Considerable Use'. Fig. 11 presents pilots' perceptions of the frequency with which they overrode or corrected CIM's actions. The average over the CIM behaviors fell between 'Seldom' and 'Now and Then' with symbol selection capabilities performing notably better.

With regards to the less observable Crew Intent Estimation behavior, pilots believed it was fairly accurate at recognizing crew intent to do "Perform Actions on Contact" (average rating 4.15 ≈ 'Frequently' triggered when crew

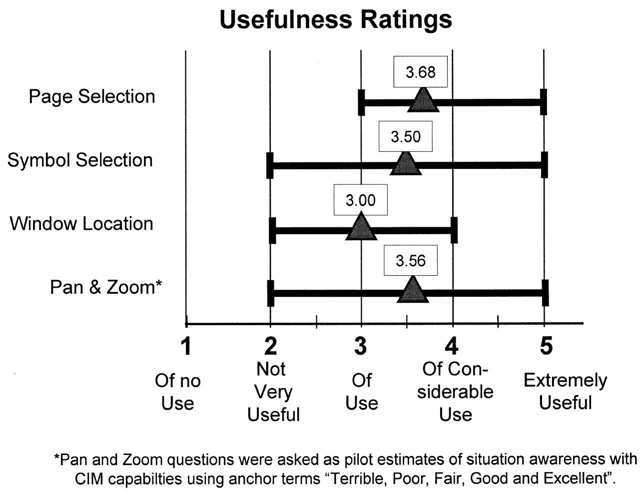

Fig. 10. Pilots' average ratings and ranges for the usefulness of the four CIM behaviors.

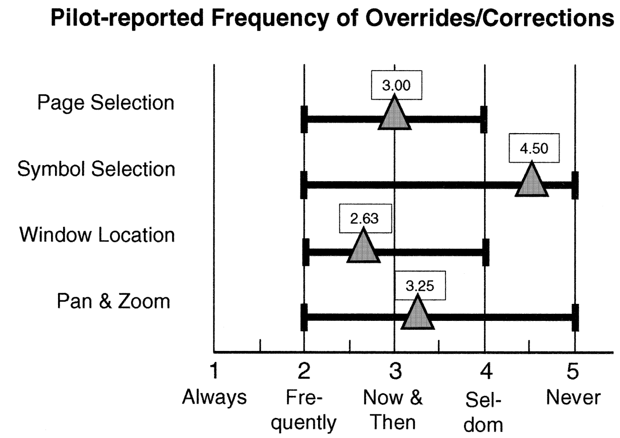

Fig. 11. Pilots' average reported perceived frequency of overrides and corrections to the four CIM behaviors.

intent or mission context made it appropriate). But this came at the cost of false positives (average rating 2.40 ≈ CDAS 'Seldom' or 'Now and Then' triggered "Perform Actions on Contact" when the pilot intended to continue past threats). In spite of these perceived occasional inaccuracies, and in spite of some pilot complaints about inadequate training in their use, most pilots found the inclusion of the LED Task Awareness display 'Of Considerable Use', as shown in Table 2.

Fig. 12 shows pilot ratings of CIM as a whole. CIM was seen as 'Frequently' providing the right information at the right time and, of critical importance to our subjects, was seen as almost always predictable in its behaviors.

Finally, while CIM cannot claim credit for all of the benefits provided by CDAS as a whole, it is difficult for both pilots and experimenters to parcel out some of these benefits. Table 3 presents a comparison of pilot ratings of their effectiveness over four high-level mission tasks with CDAS versus with the AMEP alone. On the average, pilots found themselves to be more than half a point more effective (12.5% of the scale length) with CDAS than without.

CDAS also produced overall benefits relative to AMEP in one other critical area. Using NASA's subjective taskload estimating technique, TLX, measures of subjective workload were collected at the end of each part- and full-mission trial. Perceived workload levels were consistently higher for AMEP conditions than for CDAS conditions (57 points versus 46 points). This difference was significant in an Analysis of Variance [$F(1,6) = 11.524,\ p < 0.05$]. There were no significant differences between pilots and copilots and the interaction effects were not significant.

Table 2
Perceived usefulness of the LED Task Awareness display (where 4.0 = 'of considerable use' and 5.0 = 'extremely useful')

Mission task	4.4
Pilot task	4.3
Copilot task	4.3
Associate task	4.0

Furthermore, separate ANOVAs were conducted for each of the six TLX subscale ratings to determine CDAS' contributions to overall workload reduction. As can be seen in Table 4, the reduced workload for the CDAS configuration is apparent in the mental, physical, and temporal demand and effort subscales. There is also a marginal finding for the frustration subscale ($p > 0.07$). Means in all cases indicate that CDAS provides a workload benefit to the pilot in these cases. Examination of the perceived performance ratings, however, shows no effect of technology. This may indicate that pilots use a different subjective criterion in rating their own performance, possibly judging it based on how well they felt they should have done in the given context, which would include the cockpit configuration.

4. Conclusions

Although we will have a more complete picture when the objective performance data have been evaluated, the subjective pilot responses described above suggest that the CIM behaviors we identified and implemented are generally meeting mission expectations, contributing to perceived

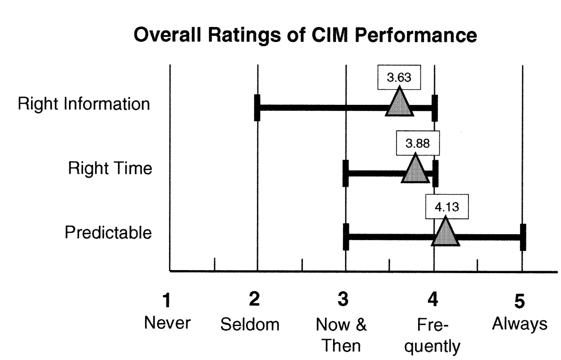

Fig. 12. Pilot's average overall ratings of the CIM.

Table 3
Perceived effectiveness in different mission tasks with CDAS and AMEP alone (where 3.0 = 'Fair', 4.0 = 'Good' and 5.0 = 'Excellent')

Average rating	AMEP	CDAS
Zone reconnaissance	3.75	3.88
Area reconnaissance	3.75	4.25
Deliberate attack	4.13	4.75
Change to attack	3.63	4.63

pilot effectiveness, reducing workload and are gaining pilot acceptance. It is worth noting that perfection in aiding and tracking pilot intent is not a prerequisite to the levels of acceptance we have gained. Pilots rated the CIM behaviors 'of considerable use' and said that CIM 'frequently' provided that right information at the right time in spite of perceived false positives and 'now and then' having to override or correct CIM's behaviors. High degrees of predictability and the addition of a highly regarded (if simple) Task Awareness and Crew Coordination Display may have contributed to pilots' willingness to tolerate these occasional 'mistakes' on CIM's part.

Areas for CIM improvement identified during this evaluation, include: (1) more predictable and accurate CIM window locations; (2) finer CIM control of the digital map pan and zoom selections; and (3) improved inhibits and/or improved intent discriminability on the "Actions on Contact" CIM behaviors when performing a deliberate attack. These modifications have been made for the RPA Flight Demonstration.

As a final indication of pilot acceptance of the CIM behaviors, when the pilots conducted each of the full-mission simulations using CDAS, pilots were given the option of turning off any or all of the CIM behaviors via a TAILOR page available both before and throughout the full mission trials. Nevertheless, all eight pilots chose to leave all CIM behaviors turned on throughout their full mission trials—a sign of trust in, and perceived benefit from, CIM's management of the displays in response to the changing mission context.

The RPA CIM is producing a range of reliable, predictable, useful cockpit interface management behaviors. The fact that pilots rank CIM behaviors highly and choose to use them even in their full-mission tests is an indicator, albeit preliminary, that CIM's IUI functions will be useful and will contribute to mission performance. As the RPA program moves through its flight demonstration, CIM behaviors will continue to be refined and evaluated, but these results give us reason to believe that they will be one of the core benefits provided by the RPA Cognitive Decision Aiding System as a whole.

Table 4
Analysis of the TLX subjective workload subscale ratings

TLX subscale	AMEP mean	CDAS mean	F-value (df: 1,6)
Mental demand	61.77	46.25	10.487[a]
Physical demand	54.48	40.31	12.042[a]
Temporal demand	62.08	45.73	14.061[b]
Perceived performance	35.00	42.08	2.429
Effort	62.60	48.54	20.470[b]
Frustration	52.81	45.63	4.961

[a] $p < 0.05$.
[b] $p < 0.01$.

Acknowledgements

This work was supported by the US Army AATD, contract number DAAJ02-93-C-0008. The authors would like to thank Harry Funk, Mark Ringer, Denny Rother, Erik Steinmetz, Ruben Yanas, Bernie McBryan and Ray Higgins for their assistance in performing the work reported here.

References

[1] S. Banks, C. Lizza, Pilot's associate: a cooperative knowledge-based system application, IEEE Expert June (1991) 18–29.
[2] F. Colucci, Rotorcraft pilot's associate update: the army's largest science and technology program, Vertiflite March/April (1995) 16–20.
[3] V. Riley, A general model of mixed-initiative human–machine systems, Proceedings of the Human Factors and Ergonomics Society Annual Meeting, Denver, CO, October 1989, pp. 124–128.
[4] T. Sheridan, Supervisory control, in: G. Salvendy (Ed.), Handbook of Human Factors, Wiley, New York, 1987, pp. 1244–1268.
[5] M. Frank, P. Szekely, Adaptive forms: an interaction paradigm for entering structured data, Proceedings of the ACM International Conference on Intelligent User Interfaces, San Francisco, CA, 6–9 January 1998, pp. 153–160.
[6] M. Maybury, Intelligent Multimedia Interfaces, AAAI Press/MIT Press, Menlo Park, CA, 1993.
[7] C.M. Mitchell, D.S. Saisi, Use of model-based qualitative icons and adaptive windows in workstations for supervisory control systems, IEEE Transactions on Systems, Man and Cybernetics 17 (4) (1999) 594–607.
[8] R. Oppermann, Adaptive User Support, Lawrence Erlbaum, Hillsdale, NJ, 1994.
[9] C. Miller, Bridging the information transfer gap: Measuring goodness of information "fit", to appear in Journal of Visual Languages and Computing 10 (3) (1999) in press.
[10] C. Miller, H. Funk, Knowledge requirements for information management; a rotorcraft pilot's associate example, in: M. Mouloua, J. Koonce (Eds.), Human–Automation Interaction: Research and Practice, Lawrence Erlbaum, Hillsdale, NJ, 1997.
[11] P. Jones, G. Chin, Varieties of representational forms: towards designing an information infrastructure for an organization, To appear in Proceedings of the 42nd Annual Meeting of the Human Factors and Ergonomics Society, Chicago, IL, 5–8 October 1998.
[12] K. Keller, K. Stanley, Domain specifies software design for decision aiding, Proceedings of the AAAI Workshop on Automating Software Design, NASA Ames Research Center, July 1992, pp. 86–92.
[13] R. Andes, Assuring human-centeredness in intelligent rotorcraft cockpits: using crew intent estimation to coordinate RPA functions. Available at www.asinc.com/papers/ahs97fnl.html. 1997.
[14] C. Wickens, Attention, time-sharing and workload, in: C. Wickens (Ed.), Engineering Psychology and Human Performance, Charles E. Merrill Publications, Columbus, OH, 1984, pp. 291–334.

[15] R. North, V. Riley, W/INDEX: a predictive model of operator workload, Applications of Human Performance Models to System Design, Plenum Press, New York, 1988.
[16] J. Greeno, Conceptual entities, in: D. Gentner, A. Stevens (Eds.), Mental Models, Erlbaum, Hillsdale NJ, 1983.
[17] S. Casner, A task-analytic approach to the automated design of graphic presentations, ACM Transactions on Graphics 10 (2) (1991) 111–151.
[18] P. Johnson, S. Wilson, H. Johnson, Designing user interfaces from analyses of users' tasks, Proceedings of the Conference on Human Factors in Computing Systems—CHI'97, Atlanta, GA, March 1997, pp. 160–161.
[19] N. Geddes, J. Hammer, Automatic display management using dynamic plans and events, Proceedings of the 6th Aviation Psychology Symposium, Columbus, OH, May 1991.
[20] C. Howard, J. Hammer, N. Geddes, Display management in a pilot's associate, Proceedings of the Aerospace Applications of AI Conference, Dayton, OH, 1989, pp. 339–348.
[21] T. Saaty, A scaling method for priorities in hierarchical structures, Journal of Mathematical Psychology 15 (1977) 234–281.
[22] S.G. Charlton (Ed.), Operational Effectiveness Questionnaire Handbook, Subjective Assessment Working Group, US Air Force Operational Test and Evaluation Center, Technical Paper 7.1, July, 1989.