A Literature Review on Operator Interface Technologies for Network Enabled Operational Environments Using Complex System Analysis

Contract Report

Dr. Haibin Zhu

Nipissing University

100 College Dr.,

North Bay, ON P1B 8L7

Dr. Ming Hou

Defence R&D Canada – Toronto

1133 Sheppard Avenue West

Toronto, ON

M3M 3B9

Defence R & D Canada – Toronto

Contract Report 2009-051

Contract Number: W7711-083931/001/TOR

2009-05-30

Abstract

A literature review was conducted to examine existing and potential advanced interface technologies for supervisory control of multiple heterogeneous assets (e.g., UAV swarming) in a NEO environment. These technologies include behavioural-based interface design approaches, physiological-based interface design approaches, and multi-agent interface design and implementation methodologies. The emphasis was on adaptive interfaces and intelligent agent system technologies. An analysis was conducted to compare differences between requirements of a NEO complex environment and currently available technologies. A review on design principles and frameworks for synthetic complex systems was also performed. The results were summarized with pros and cons of different technologies for interface design purposes.

Résumé

Une analyse documentaire a été menée pour examiner les technologies d'interface de pointe existantes et potentielles permettant le contrôle de surveillance de nombreux biens hétérogènes (p. ex. groupes de VAT) dans un environnement d'opérations réseaucentriques. Ces technologies comprennent des approches de conception d'interface fondées sur le comportement et sur la psychologie, ainsi que des méthodes de conception et de mise en œuvre d'interfaces multi-agents. On a mis l'accent sur des interfaces adaptatives et sur les technologies de systèmes à agent intelligent, et on a analysé les différences entre les exigences d'un environnement d'opérations réseaucentriques (NEO) complexe et les technologies actuelles fournies. De plus, on a examiné les cadres et les principes de conception pour les systèmes complexes synthétiques. On a résumé les résultats avec les avantages et les désavantages de différentes technologies à des fins de conception d'interfaces.

Executive Summary

Network Enabled Operation (NEO) is a fundamental paradigm shift from platform-centric operations (PCO). NEO is defined as an information superiority-enabled concept of operations that generates increased combat power by networking sensors, decision makers, and operators. This entails shared situation awareness, increased speed of command, higher tempo of operations, increased survivability, and a degree of self-synchronization. NEO translates information superiority into operational power by effectively linking knowledgeable entities in the battle space. Consequently, as the number of entities, or systems, in the battle space increases, the resulting NEO system-of-systems will likely exhibit more complex emergent behaviour which needs to be properly addressed as we move from PCO to NEO.

Emergent behaviour occurs when a group of entities interacting with each other exhibits behaviour that cannot be readily predicted. It is termed global emergent behaviour if a small subset of the entities when separated from the group will not have the same behaviour as they would have within the group. This is a fundamental characteristic of complex systems and the possibility of applying complex systems analysis techniques for the solution of military and public security problems has recently begun to be recognized. It is a well-known occurrence that large-scale operations fail because the complexity of the interactions in the operations is not well understood.

To explore and understand complex systems with the context of network-centric warfare, Defence Research & Development Canada (DRDC) has started a project entitled "Synthetic Environment Experimentation for NEO of Heterogeneous Assets Using Complex System Analysis". The purpose of this project is to develop effective supervisory control concepts of coalition heterogeneous assets that will aid military and public security operations against asymmetric threats. In order to do so, a networked synthetic environment needs to be generated using a NEO approach which combines complex system principles with artificial intelligence methodologies and intelligent adaptive operator interface technologies. Thus, an operator interface needs be developed in the complex and synthetic environment to provide decision making aids and improve situation awareness when operators are overloaded with massive information.

This research surveyed and synthesized the body of knowledge on principles and best practices of using complex system analysis approach for operator interface technologies to aid NEO. To integrate technologies into CF synthetic environments, this work also investigated most suitable integration components and designed an implementation framework. This report summarized recently developed theories, frameworks, and best practices for Operator Interface Technologies for complex systems. By comparing their advantages and disadvantages, it provides a solid base to synthesize guidance to design operator interfaces for network enabled operation environments.

Table of Contents

1. Introduction

Automation is defined as the delegation of tasks to machine or computer systems, thus reducing procedural load and freeing operators from vigilance over routine and tedious tasks (Berken, et al. 1991). It can also be defined as a device or system that accomplishes (partially or fully) a function that was previously, or conceivably could be, carried out (partially or fully) by a human operator (Parasuraman, et al. 2000). The classic aim of automation is to replace human manual control, planning and problem-solving by automatic devices and computers. However, even highly automated systems need human beings for supervision, adjustment, maintenance, expansion and improvement. Therefore, both systems and human factors are important (Bainbridge, 1983).

With advances in robotics, artificial intelligence, and human-machine interactions, more and more automatic systems have been applied broadly. One typical technology that has been introduced and explored actively in the Canadian Forces (CF) is the Uninhabited Aerial Vehicle (UAV). Due to its rapid deplorability, increased communication and joint operations capability, and other revolutionary potentials in battle fields, UAV systems became an effective tool for CF's command, control, communications, computers, intelligence, surveillance and reconnaissance (C4ISR) (Edwards, 2004; Gil, et al. 2008; Ruff, et al. 2002). With a greater use of intelligence in unmanned vehicles in battle fields, these systems can supplement traditional manned forces as force multipliers. US Congress has mandated that up to one third of future military systems are unmanned by 2015 (National Research Council 2003). A key benefit of UAVs is their ability to carry out missions that would endanger human life (e.g., suppression of enemy air defence or reconnaissance in contaminated areas) (Hou, et al. 2007).

Although the UAV has characteristics of an autonomous system, it works with a remote control station that is separated from the vehicle and supervised by human operators (Gil, et al. 2008; Manning, et al. 2004; Schulte, et al. 2009; Wilson & Russell, 2007). The functions of the control station include (1) receiving telemetry data from the UAV through a wireless modem, (2) processing these data, (3) displaying the UAV's status, and (4) supervising its navigation at waypoints. The control station can be put on the ground or on another mobile carrier (e.g., ground vehicle, airborne platform, or vessel). A control system has a variety of features and settings and thus requires an operator to be aware of everything and be able to exercise control continuously (Powers, et al. 1973; Rouse, et al. 1988). Obviously, such knowledge-intensive UAV control can generate extremely high workload for an operator. In addition, the situation can be exacerbated by system-time delays and reduce the operator's situational awareness due to the extensive information processing demand. The RQ –1 Predator UAV mishap partially resulted from losing situational awareness of a fuel problem and finally loss of engine power. Two UAV pilots became too focused on the rarely encountered severe weather conditions, lost control of the UAV and were unable to recover (Rogers, 1999; Manning, et al. 2004).

The loss of the Predator UAV indicates that current UAV control stations/interfaces cannot provide adequate information about the system states as well as operators' needs due to the nature of complex and dynamic information management for the UAV control. In order to increase operators' situational awareness (Manning, et al. 2004; Guyot & Honiden, 2006) and thus provide the right information to the right person in the right format at the right time, the alternative is to employ automated software assistants to take over some of the tasks rather than having operators do everything themselves. These automation assistants can act as agents to acquire knowledge about the mission goals, the operator's goals and states, as well as the environmental states. An agent may perform a function automatically, ask permission to act, provide information, ask for clarification, or relinquish control to operators. Thus, operators can delegate some responsibilities to these automatic agents and focus exclusively on the crucial tasks that need human oversight. Agent systems have been widely used in software engineering and artificial intelligence (AI) fields and an agent-based approach is well suited to assist operators in complex and dynamic systems. However, there is still no evidence on how much agents can help operators reduce workload and increase situational awareness and effectiveness. In addition, how operators react to these various levels of agents/automation to enhance overall human-machine performance needs to be investigated.

The aim of this paper is to develop an agent-based framework to guide intelligent adaptive interface design for complex systems (e.g., UAV control station). Through the optimization of operator-agent interaction, a multiple agent-based interface is expected to increase operators' situational awareness and reduce workload and thus supports reduced manning and enhanced performance in complex military systems. Three specific aspects of this research are identified and examined in detail. First, some commonly used terms are properly defined to clarify potential confusion with similar terms used in other research areas. The challenges in intelligent adaptive interface design are also identified. Second, relevant research and empirical investigations are reviewed in terms of concepts and principles for advanced agent-based interface design. Finally, a framework is proposed as guidance for interface design based on intelligent software agent technology, a user-oriented approach, and the proactive use of automation. Through adaptation, allocation and automation delegation (Miller, 2005; Miller & Parasuraman, 2007), it is argued that a multiple-agent based architecture increases operators' situational awareness and facilitates the processes of knowledge acquisition, attention, reasoning, and decision-making. The framework also provides guidelines for further empirical investigations on the effect of task complexity, operator workload, and interface intelligence.

2. Method

The approach to the literature review had three steps:

1) *Literature search*.

Following the development of appropriate search criteria approved by the Scientific Authority (SA) (e.g., keywords, authors, organizations), the CAE Professional Services (Canada) (CAE PS) team searched scientific, defence (e.g., Canada, United Kingdom, United States defence and NATO reports), government (e.g., DRDC Toronto reports and other documentation provided by the SA) and internet-based sources such as Google Scholar for literature pertaining to intelligent adaptive systems;

2) *Reduce and collate literature.*

In collaboration with the SA, the CAE PS team developed an EndNote x2 database built to classify references by Levels of technical reports, conference papers, journal papers, book chapters, and books. The results of the literature search were collated and reduced according to appropriate selection criteria. The literature search is confined by Network Enabled environments and Complex System Analysis. With the criteria, any abstract model with many simplifying assumptions is ignored.

Recommendations or guidelines developed from sources involving experimental studies within the military domain that have been subject to critical peer review were given more prominence in the report than those that have been developed from more generic, conceptual sources, or that were subject to little or no peer review. All references were earmarked for inclusion into or exclusion from the final review, and classified according to area (i.e., conceptual frameworks, analytical techniques, design principles, and physiological/behaviour-based adaptation).

3) *Development of reporting structure*. From the collated literature, a structure was developed for reporting findings in conjunction with the SA; and,

The literature review was looked at the following:

- *Frameworks.* This document reviews theoretical frameworks, such as those adopted by the Cognitive Cockpit, Pilot's Associate programmes, and the DRDC UAV Interface Design project. The literature research reviewed, selected and described conceptual frameworks for designing IASs, including, but not limited to, those described above. In addition, the review also highlighted important similarities and differences, and advantages and disadvantages, between the theoretical approaches;
- *Agent-based Design Principles*. This document reviews approaches to understanding and aiding human interaction in real-world complex systems from a multi-agent perspective. The literature research reviewed, selected and described issues relevant to the understanding and interaction between human and machine agents in the design of IASs (e.g., team work, organisation); and,
- *Operator-state Monitoring Approaches.* This document reviews techniques for the analysis of the psychological, physiological and behavioural states of an operator in order to provide information about the objective and subjective state of an operator within a mission context. As with knowledge of the external context, information about the internal (i.e., operator) context provides the basis for an intelligent adaptation of the automation and/or interface to support the operator to achieve system goals. The literature research reviewed technologies for designing behaviour-based and physiological-based interface systems, compared differences between behaviour-based and physiological-based techniques and also identified the benefits of combining the two techniques.

3. Intelligent Adaptive Interface and Adaptive Intelligent Agents

The maturation of computer, communication, and other advanced technologies has led to increase automation in nearly all aspects of society. For instance, through the use of AI technology, software systems are gaining the ability to reason and make decision on their own. This trend has triggered a shift in the human operator's role from largely perceptual/motor (e.g., controlling a vehicle by direct manipulating a stick or wheel) to more cognitive processes (e.g., monitoring, reasoning, judging, and managing automated or semi-automated system operations). Consequently, the design of effective and efficient human-machine interfaces becomes ever more critical to overall system performance as both operators and the system have to adapt the role shifts. However, the requirements imposed by human-computer interaction (HCI) with such systems exceed the capabilities of conventional interfaces that often fail to reflect their users' tasks, goals, plans, and problem domain properly (Bowan & Hodges, 2000; Sukaviriya 1993; Sutcliffe, et al. 2006). In addition, there is a lack of clarity about what comprises an interface that can aid users in decision-making and very little guidance on how to evaluate its effectiveness. This is especially true for complex human-machine systems (Jones, et al. 1995) such as a UAV control station where information flow is dynamic, operators have excessive workload, and situational awareness is crucial to accomplish the mission. This section starts by defining two terms: adaptive intelligent agent (AIA) and intelligent adaptive interface (IAI) in the human-machine interaction domain, and the differences between human-machine interaction and operator-agent interaction when using agent technology as a means for the interface design. The challenges for the design of IAIs that incorporate a variety of AIAs are also discussed.

3.1 Adaptive Intelligent Agent (AIA)

Improvements in AI techniques and the growing need for better interface metaphors have led to two converging and often conflated areas of research on agents (Lieberman, 2002). Generally, an agent refers to a representative authorized to act for another. A travel agent makes an individual's vacation plans, and a stockbroker selects an opportune time to purchase a stock for customers. In software technology, the use of the word "agent" denotes that they can take action on behalf of a user. Software agents became an active research area since the early 1990s when the technology has shown promise as an effective computing paradigm to implement users' delegations and make the interface an active partner to the users (Shneiderman & Maes, 1997; Grossklags & Schmidt, 2006).

The term "agent" has been used differently in various research communities, but definitions of agents in the user interface domain are most relevant here, sometimes called "interface agent (Baylor, et al. 2006; Gómez-Sanz & Garijo, 2000; Laurel, 1990)". Although there are still multiple definitions for this (Bradshaw 1997), there is a special kind of agent that share several common characteristics. As Finin, et al. (1997) and Wooldridge (1999, 2002) identified, their most prevalent characteristics are adaptation (or reactivity to the changes in users and environments), autonomy (for taking certain tasks from users), and cooperation (or social ability among agents themselves), as illustrated in Figure 1. Thus, these three characteristics become the defining attributes of adaptive intelligent agent (AIA). In the HCI domain, an AIA can be defined as:

A personification of computer code and algorithms so as to mimic human perception, cognition, and behaviour to some extent, which can cooperate with other agents, automatically take actions on the user's behalf, and adapt to the changes in the user and environment.

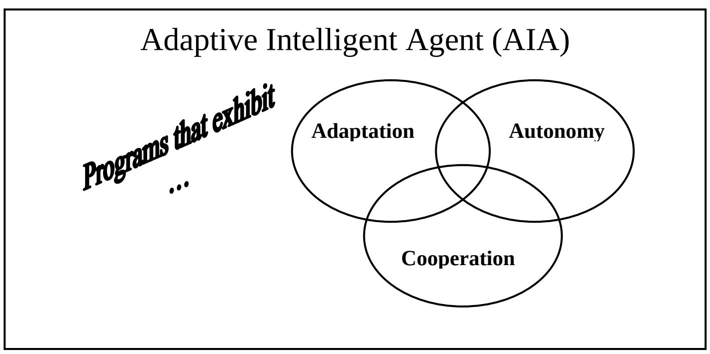

Figure 1. Characteristics of an adaptive intelligent agent (Finn, et al. 1997).

Therefore, an AIA is a software program that exhibits adaptation, autonomy, and cooperation in accomplishing a task for the user. Although there is argument on the concept of adaptation (Burgos, et al. 2007), an AIA should be able to adapt autonomously to facilitate the cooperation between operators and the interface of the system. An AIA can be regarded as a human operator's partner that can autonomously take over certain tasks from operators and adapt itself to the changes in operators and environments.

According to Weld (1995), an intelligent agent is referred to as "an entity capable of autonomous goal-oriented behavior in some environments, often in the service of larger-scale goals external to itself." Haynes, et al. (2008) define intelligent agents as "software programs designed to act autonomously and adaptively to achieve goals defined by their human developers or runtime users (the latter can be other intelligent agents)." In the context of UAV control, AIAs embedded in an operator interface are well suited to taking over some tasks (e.g., monitoring fuel state or altitude level) delegated from human operators, serving to support the goals of optimizing operator-agent interaction in a complex and dynamic environment to maximize overall performance.

It is noted that using this definition permits a broad range of applications to be classified as AIAs but excludes many “normal” range of computer programs. Based on these characteristics, for example, a standard macro would not be classified as an AIA. A macro is usually a small program embedded within a larger application. As such, macros implement or automate a task for the user, but they are usually input- and environment-dependent. Any deviation from the initial inputs or changes to the environment can cause the macro to fail.

3.2 Intelligent Adaptive Interface (IAI)

human-computer interfaces have difficulties in conveying and understanding a large amount of information required and processed by complex task structures for advanced applications such as multiple UAV control. It is natural for an operator to ask some “intelligence” for help to reduce the information processing workload (if some of the advanced technologies such as AI can be applied in the interface design). Such machine intelligence embedded in user interfaces aims to improve the efficiency, effectiveness, and naturalness of human-machine interaction by representing, reasoning, and acting on the models of the user, domain, task, discourse, and media (Maybury, 1999; Banbury, et al. 2007).

The term “intelligent adaptive interface” (IAI) proposed here is adopted from intelligent interface concept (Jaquero, et al. 2009; Khoshnevis and Austin, 1987; Lipa, 1999; McTear, 1993; McTear, 2000; Meyer, et al. 1993; Yoon and Kim, 1996). It is intended to capture the wide range of issues and methodologies in which some intelligence is applied to user interfaces. Originally, intelligent interfaces became known as adaptive user interfaces (Abascal, et al., 2008; Benyon and Murry, 1988, 1993; Browne, et al., 1990; Hancock, 1988; Innocent, 1982; Langley, 1999; Liu, et al. 2003; Maat & Pantic 2007; Mitrovic, et al., 2007; Norcio & Stanley 1989; Tyler 1990). In the context of human-machine interaction, Powers, et al. (1960) and Sternberg (1996) defined intelligence as the ability to perceive, understand, reason, and infer the relevant changes of environments and adapt accordingly. Tomlinson, et al. (2007) define adaptive interfaces as interfaces that seek to predict what sorts of features would be desirable and that are customizable by users. Thus, an intelligent interface should be able to tailor some of its parameters to certain prescribed specifications or convert itself to adjust to changing circumstances, requirements or needs. In other words, it has adaptive capabilities to the user and environment. Therefore, an IAI can be defined as:

An operator interface that dynamically changes the display and/or control characteristics of human-machine systems to adaptively react to external events (mission and operator states) in real time. (Hou, et al. 2007).

A typical IAI is driven by software agents (automation) that intelligently aid the decision-making and action requirements of operators under different levels of workload and task complexity by presenting the right information or action sequence proposals or performing actions in the correct format at the right time. Clearly, the emphasis in this definition is on both intelligence and adaptation.

3.3 Difference between Human-Human Interaction, Human-Machine Interaction, and Operator-Agent Interaction

The focus of IAI technology proposed here is the interaction between operators and agents for better overall performance. To analyze operator-agent interaction, understanding human-machine and human-human interactions should serve as a starting point because many human factor issues in these areas have been extensively studied for decades and should be effectively applied in an IAI design process.

First of all, the differences in their definitions should be reviewed as below.

- Human-Machine Interaction: interplay between a human and a machine that is more than just a computer, operating in a fairly well-known environment.
- Human-Computer Interaction: interaction between a human and a computer typically in a sedentary environment.
- Human-Human Interaction: communication between humans where beliefs and trust are of significant importance as the context changes.
- Operator-Agent Interaction: exchanges between a human and an intelligent agent system as the context changes.

Table 1. Differences between human-machine and operator-agent interactions

Human-Machine Interaction (HMI)	**Operator-Agent Interaction (OAI)**
Plans, actions and system states are known within limits.	Plans and actions are not known a priori, and may produce unexpected system states.
It is specific, systematic, and often associated with Standard Operating Procedures.	It is fuzzy and there may be many means to achieve the same end.
The human has beliefs (assumptions) about the machine and task. The machine design takes into consideration certain assumptions about the human and task.	The operator has beliefs about the agent and task. The agent may have the equivalent beliefs about the operator and task.
Typically, trust is binary - the machine works or does not.	Trust must be built over time because there are few definite ways to judge whether the agent is performing well.
Typically, there are two levels of autonomy – completely manual or completely automatic.	There are multiple levels of autonomy.
The human knows how the system will process/display information.	The operator does not necessarily know how the system will process/display information.
The context is typically static or well-defined.	The context is dynamic and sometimes unknown a priori.

From the above definitions, both human-machine interaction and human-computer interaction typically have static or well-defined context based on the assumptions, rules, and procedures. The difference between them is subtle except that the former may involve more direct manipulations than the latter. They are both different from human-human interaction and operator-agent interaction as the latter is both dynamic and its context is sometimes unknown a priori. Although there are some overlaps between human-machine interaction and operator-agent interaction as agents are usually embedded in a machine, there are still some major differences between them as highlighted in Table 1.

Because of the differences, existing human-machine interface design standards are probably still relevant but not sufficient for agent-based IAI system design. If designers are using current standards to design intelligent agent systems, this may be one reason that users find these systems frustrating. A fresh look at operator-agent interaction may capture new principles or guidelines that do not appear in current standards. Thus, there is a strong need to develop guidelines to meet IAI design challenges.

Second, since the AIAs aim to help users by automatically taking over some tasks and adapting themselves to the changes in users and the IAI, they should be more human-like and resemble some human behaviors for effective interaction. Thus, operator-agent interaction should adopt some human behavior: by its nature, physiological attributes (eye, ears, fingers), intellectual characteristics (e.g., capacity, recognition, learning, and decision), knowledge basis (knowing the environment, the system, the task, the user, etc.) and psychological states (e.g., concentration, vigilance, fatigue, and patience), as illustrated in Table 2.

Table 2. Behavior of human-human interaction

Behavior	**Component**
Observation/Perception	Aural, verbal, visual, tactile, force
Communication	Ears, mouth, eyes, fingers, hands
Cognitive Process	Learning, understanding, reasoning, reference, trust
Adaptation	Influencing beliefs, changing behavior
Collaboration	Reduce workload and improve situation awareness and performance

An example of an agent paradigm resembling cognitive and adaptation processes of human behavior is the belief-desire-intention (BDI) approach (Bratman, *et al.* 1988; Georgeff, *et al.* 2000; Jarvis, *et al.* 2005; Sudeikat, *et al.* 2007). An agent's beliefs refer to the knowledge the agent has about the world, which may be incomplete or incorrect. An agent's desires correspond to the tasks allocated to it. In other words, goals in the system are required to be logically consistent. However, an agent will not be able to achieve all the system goals even if they are consistent. An agent must thus fix on some subsets of available desires and commit resources to achieving them. These chosen desires are then defined as intentions. An agent will typically continue to achieve an intention until the intention is believed either satisfied or no longer available (Cohen & Levesque 1990). The key data structure in a BDI model is a plan library, which is a set of plans for an agent. This set of plans (recipes) specifies courses of action that an agent may take to achieve its intentions. An agent's plan library represents its procedural knowledge about how to produce system states. This approach is referred to as well-known agent architecture, Procedural Reasoning System (PRS) (Georgeff & Lansky, 1987), in agent technology although it is not a complete human behavior model from a psychological perspective.

The BDI agent model only resembles cognitive and adaptation processes of human behavior. There should be more human-like agents to imitate other human behavior as listed in Table II as well. Only when a number of these agents are effectively implemented and deployed in IAIs can the operator-agent interaction promise better human-machine interaction with more benefits including:

- More effective interaction – doing the right thing at the right time, tailoring the content and form of the interaction to the context of users, tasks, system, and communications;
- More efficient interaction – enabling more rapid task completion with less work;
- More natural interaction – supporting spoken, written, and gestural interaction, ideally as if interacting with a human interlocutor.

Given the knowledge of the difference between human-machine and operator-agent interactions, the most effective way to design an effective and efficient interface is to incorporate many proactive and personalized AIAs into an IAI. AIAs can act as partners and take some responsibilities delegated from operators for optimizing operator-agent interaction naturally and thus to maximize overall system performance.

4. Current State of IAI Design

The area of IAIs covers a variety of topics concerned with the application of AI and knowledge-based techniques to issues of human-computer interaction. The main issues addressed by IAI research are a) making interaction clearer and more efficient; b) supporting users' tasks, goals, and plans; c) presenting information effectively; and d) designing and implementing interfaces effectively (Hook, 2000). To address these issues, this section starts with a brief introduction to several IAI design approaches, two empirical studies, and the design challenges.

4.1 Agent-based Interface Design Approaches

For most complex systems (Rouse, et al. 1987; Xiao, et al. 1997) (e.g., multiple UAV control stations), the interface development requires high skill level and massive time commitment. Thus, the techniques for interface development were originally developed as interface design and development environments (Maybury, 1999). These environments are essentially software components such as windows, menus, and dialogue boxes, e.g., X window, Open-GL, and Visual Basic. These toolkits support design consistency and enhance programming productivity via code reuse. Unfortunately, they frequently mix interface code with application code. To solve this problem, model-based interfaces separate applications into more layers (i.e., application actions, dialogue control, specifications of presentation and behavior, and primitive toolkit objects composed by specifications) to support more declarative development. Model-based systems (e.g., MAYA™) can draw upon automated input analysis and output generation techniques. In contrast to interface software toolkits, model-based interface development environments promise automated design critique, refinement and implementation. However, these design environments neither emphasize human-computer interaction for the outcome interface, nor provide enough means to facilitate the interaction. On the other hand, an agent-based interface design approach emphasizes the use of AI techniques embedded in the interface to create software that performs information filtering and other autonomous tasks for users. The essential function of embedded automation agents is to act as effective bridges between a user's goals and the computer's capabilities. As the interface metaphor, an agent is used to make the interface more intuitive and to encourage types of interactions that might be difficult to evoke with an interface developed in a model-based environment. There are many agent-based techniques to help with interface design in systems engineering and AI fields. Among them, two design approaches have the potential to provide means to facilitate operator-interface interaction: Common Knowledge Acquisition and Design Structuring (CommonKADS) and Explicit Models Design (EMD).

CommonKADS is a methodology promoted by Schreiber, et al. (2000) for developing a knowledge-based system. It provides guidelines on a methodology to analyze and design before any code is written. It has a formalized representation of knowledge and an associated inference mechanism with the capabilities of human activities: diagnosis (Yoon & Hammer, 1988), planning, and design. A Multi-Agent System extension of the CommonKADS methodology (MAS-CommonKADS) has been proposed by Iglesias, et al. (1996). It was developed to add specific agent-related constructs, including those relating to: a) inter-agent communication; b) the division of tasks among individual agents; and c) the implications for implementation of multi-agent systems.

Although these two methodologies can offer a framework for approaching the design and implementation of an agent-based system (e.g., an IAI), they do not provide a means of identifying and subdividing the knowledge required by IAI systems. In addition, they are inflexible to represent temporal relationships and constraints among tasks, inferences, and data flow because they use the Unified Modeling Language (UML) to represent those elements.

Integrated Computer-Aided Manufacturing (ICAM) Definition (IDEF) was developed by the US Air Force (Lydiard, 1995; Mayer, et al. 1995) to look at methods for analyzing and improving manufacturing operations. Its elaboration language allows symbolic representations of complex constraints that cannot be depicted using the schematic language, e.g., UML, alone. Thus, it permits flexible modeling of temporal concepts and formal logical representations of process constraints and allows precise specification of event timings and durations. Consequently, it can solve the above mentioned problem of CommonKADS. Therefore, integrating the above approaches and combining the strengths of the individual components will provide an opportunity to form a comprehensive approach for IAI system design.

Another method is the EMD that promises to offer a means of subdividing the content of the CommonKADS knowledge into five distinct and interacting components: task, user, system, dialogue, and world (Edwards and Hendy, 2000). By making explicit the knowledge required by the IAI systems, EMD has the potential to determine the goals a user is trying to achieve, the plans for achieving those goals and how it can assist the operator most effectively. Challenges for EMD lie in how to decompose knowledge into the various models and how to co-ordinate the knowledge among the models to build effective support systems. To tackle these challenges, a technique called Hierarchical Goal Analysis (HGA) and associated Perception Control Theory (PCT) provide mechanisms for decomposing user and system goals and recognizing user plans based on observed actions (Hendy, et al. 2002). Thus, these techniques have been integrated in EMD for programmatic implementation of the method. Since the details of these techniques are beyond the scope of this paper, they will not be addressed here.

Besides this, there are other methodologies to address interface design issues such as Cognitive Task Analysis (Jonassen, et al. 1999; Schraagen, et al. 2000; Kaber, et al. 2006) and Ecological Interface Design (Burns and Hajdukiewicz, 2004; Jamieson et al., 2007; van Dam, et al. 2008; Vicente & Rasmussen, 1992). Although they are not formal design environments, they both have good sets of techniques for constructing safe and reliable interfaces based on the levels of cognitive control. However, they still need more research on their strengths and weaknesses as complete methods on operational or behavior level.

Maat, et al. (2007) proposed Gaze-X: an adaptive, affective, multimodal HCI system in standard office scenarios. This system is based on the sensing and interpretation of the human part of a computer's context. The adaptation of Gaze-X is that the interface changes as a function of the currently sensed context. This function is represented by the cases of the utilized dynamic case base, having the system's and user's states as the input and the adaptive and user-supportive changes in the interaction as the output. A usability study conducted in an office scenario indicates that Gaze-X is perceived as being effective, easy to use, and useful by novice and less experienced users, as well as being usable and affectively qualitative by all participants in the study. Gaze-X seems to be very suitable for novel and intermediate computer users but is much less so for experienced ones. As only one experienced user has participated in the present usability study, a much more elaborate survey must be conducted with many experienced users if a firm conclusion is to be made about the ways to make Gaze-X useful and appealing to experienced ones.

The ISATINE framework for user interface adaptation (López-Jaquero, et al. 2008) was initially aimed to support more than just the adaptation execution. It decomposed user interface adaptation into seven stages of adaptation: goals for adaptation, initiative, specification, application, transition, interpretation, and evaluation. The purpose of each stage was defined and could be ensured respectively by the user, interactive system, a third party, or any combination of these entities. It distinguished three forms of coordination: negotiation, delegation and transfer in the adaptation process.

In the ISATINE framework, the BDI paradigm was applied. A graph transformation system, consisting of steps of graph transformations, was developed to support the execution of the adaptation on a graphical user interface (GUI) model. The adaptation process could be initiated by either the user, system or a third-party. After initiation, a set of adaptation rules that best fit the current context of use was proposed. It just supported transitions among general adaptations, since the transitions were generated at run-time on-the-fly. This framework was built on a multi-agent architecture including agents to support each stage.

Letsu-Dake and Ntuen (2009) proposed a conceptual model for designing adaptive HCI. They introduced the living systems theory (LST) as a paradigm for HCI design that could produce interfaces with self-learning and adaptive capabilities. LST is a kind of biological model that allows designers to simulate adaptive behaviors of ‘‘intelligent’’ systems. They believe that an LST framework provides the attribute required to encourage self-organized behaviors during user interaction with computers. They build a sample interface in the domain of power system information to demonstrate the efficacy of LST as an adaptive HCI design tool. Letsu-Dake and Ntuen showed that the application of LST will not only provide a robust framework for designing HCI, but can also improve the performance of the adaptive processes of the interface agents.

4.2 User-Centered Design

User-centered or human-centered design was initiated by Norman and Draper (1986) (Karat 1997; Keinonen, 2008; Norman, 2005). “User-centered design tries to optimize the user interface around how people can, want, or need to work, rather than forcing the users to change how they work to accommodate the software developers approach” (Wikipedia, 2009).

A typical representative of this user-centered philosophy is an intelligent interface architecture proposed by Rouse, et al. (1997). First, the operator state is central to the functioning of the components of the intelligent interface. The relevant elements include: activities, awareness, intentions, resources, and performance. The other component is the interface manager that is similar to an executive's assistant who zealously guards the superiors' time and resources. The user-centered principle is used as a key component in the proposed framework for guiding interface design. However, the concept of automation used only as a back-up is not applicable for a dynamic, complex, and interactively networked system such as multiple UAV control. When dealing with a large amount of information while in complex mission environments, what an operator really needs for the control of multiple UAVs is to employ automation to help reduce workload by delegating tasks before anything goes wrong. Otherwise, it may be too late. Thus, instead of being a backup, automation should act like a partner to proactively prevent anything from going wrong. Therefore, the operator and the automation will work as a team and share the responsibilities to maximize overall performance – resembling the behavior of human-human interaction. In this way, the IAI does not act like a passive "listener" but an active partner and the operator will not be a passive observer but an active controller. With the agreements on who does what packed in advance, some tasks will neither be missed nor duplicated.

4.3 Empirical Investigations

Although the theory of IAI has been around since the early 1990's (Lamberti & Wallace, 1990), there is still a need to verify it because there were only a few empirical investigations on IAI technologies (Eklund & Sinclair, 2000). Three relevant studies are considered as applications of IAI technology and reported here.

Bennett, *et al*. (2001) conducted a preliminary study of a dynamically adaptive interface (DAI) in the domain of aviation (precision, low-level navigation). A DAI could change display or control characteristics of a system (or both) in real time. Its goal was to anticipate information needs of users and provide that information without the requirement of an explicit control input by users. In this research, three interfaces were evaluated including: standard (conventional controls and displays), candidate (alternative controls and displays), and adaptive (dynamically between the standard and candidate displays). The results indicated that significant performance advantages in the quality of route navigation were obtained with the candidate and adaptive interfaces relative to the standard one. However, there were no significant differences between the former two. The implication of this study is that adaptive interfaces have the potential to improve overall human-machine system performance if they are properly designed. But if designed improperly, adaptive interfaces have the potential to degrade system performance by (a) preventing the development of automatic processes in the operator; (b) presenting irrelevant information; or in the worst-case scenario, (c) eliminating information that is currently needed. In addition, the other implication of this research raised the issue of the dilemma for automation and adaptation using IAI technology.

To address the dilemma of automation and adaptation in applying IAI technology, the key research questions that need to be answered before any interface design are: what can and should an agent do? Who (which agent) should interact with users? When, where, why, and how should an agent do it? This is called W5 questions by Duric, *et al*. (2002) and W5+ by Maat, *et al*. (2007). Duric, *et al*. discussed adaptive intelligent HCI by integrating perceptual and cognitive modeling through both theoretical analysis and empirical investigations. Although the research has shown only the effectiveness of the tools for the interpretation of perceptual processes, including lower arm movements, facial data processing, eye-gazing tracking, and mouse gestures, other IAI technologies (i.e., behavioral and cognitive tools) were also advocated in interface design for intelligent HCI where human cognitive, perceptual, motor, and affective factors can be modeled and used in the interface to adapt the changes of users. The essence of the research was to monitor affective or emotional behavior, or non-verbal information to answer W5 and adapt the display according to user behavior. The method for interface design was more human-like in which the interface/machine or the embodied agent/automation was regarded as another human assistant who could monitor perceptual and cognitive states and understand users as partners or teammates, thus react for better collaboration with better overall results. The idea was to emphasize the team collaboration, which is true in human-human interaction (see Table 2). In human-human interaction, since everyone does not work well with everyone else, everyone has to adapt to the changes of others to achieve better overall collaboration performance.

Based on their earlier implementations of three adaptive user interfaces (AUIs): Split, Moving, and Visual Popout, Gajos, *et al*. (2006) designed two experiments to investigate how different design choices and interactions affect the success of adaptive GUIs. A Split Interface copies important functions onto a toolbar in a spatially stable manner. Users could choose either to continue using the (unmodified) original interface or to use the adaptive toolbar. A Moving Interface *moves* promoted functionality from inside popup panes onto the main toolbar, causing the remaining elements in the popup pane and the existing buttons on the toolbar to shift to make space for the promoted button. A Visual Popout Interface highlights promoted buttons in magenta. If a promoted button resides inside a popup menu, both the button invoking the popup menu and the menu item are highlighted.

Gajos, *et al*. (2006) computed an estimate of the participants' perceived benefits and costs associated with three adaptive interfaces. They used the average of the responses to the Efficiency and Performance questions as a measure of benefit. The Mental Demand, Physical Demand, Frustration, and Confusion questions were used to compute the cost. The Split Interface was found most beneficial and least costly despite having lower theoretical benefit than the Moving Interface. The Visual Popout Interface was found to confer little benefit, as expected, but participants found it very distracting and assigned it a higher cost than they had expected.

They observed that users perceive adaptations as incurring a cost. Purely mechanical properties of an adaptive interface are not a good predictor of a user's performance or satisfaction. The balance between perceived cost and benefit would be a better predictor of user acceptance.

The most important implication from these studies was the recognition that not only does computer technology need to make such novel interfaces a reality, users as well adapt to the interface that the computer presents them with. In the end, both operator and the computer must understand each other's intentions and/or motivations, provide feedback to each other as necessary, and eventually adapt to each other.

4.4 Challenges of Intelligent Adaptive Interface Design

Intelligent user interface, in other words, intelligent adaptive interface (IAI) technology has been proposed by AI and knowledge-based system design community as an effective means to overcome the problems that conventional direct manipulation interfaces cannot handle: information overflow and real-time cognitive overload. In the past decade, a number of theories were developed and tested (see the above two sections). However, there are still a number of problems that prevent this technology from creating good applications. The challenges for IAI design include the following (Hook, 2000):

- What are the universal usability principles that do not lead users' expectations astray?
- What are the reliable and cost-efficient IAI development methods?
- How and when can intelligence substantially improve the interaction?
- How do we evaluate whether the system supports users' real tasks? and
- How do we design authoring tools to enable easy development and maintenance of the intelligent parts of the system (Scalability)?

In the context of multiple UAV control in a dynamic and complex environment, the interface demands better usability principles, better ways to improve the interaction, and better tools to support the full life cycle of a human-machine system (e.g., UAV control). In addition, in a networked environment in which human operators interact with UAVs, the problem becomes more complicated during combined operations where interoperability is important and many forces use information from one UAV.

Furthermore, in terms of AIA-based operator-agent interaction and evaluation in the IAI domain, research on agent technology has increased in prominence in the systems design field, which includes the use of AIAs to express a system and system status via facial displays, multimodal communication between animated computer agents, and standards and open architectures for building agent-based multimodal interfaces. However, the key questions are: what can and should AIAs do, how should they do it, and how, when and why should they interact with users while doing it? An agent-based generic framework proposed in the following section is going to address these issues from a human factors perspective.

5. Generic Intelligent Adaptive Interface Design Framework

As indicated in the previous section, there is little research focusing on operator-agent interaction as the agent-based technology has not been adopted for designing many complex systems. In particular, from a human factors engineering perspective, a lack of theoretical developments and empirical validation studies makes many designs costly and ineffective. Many existing frameworks focus on individual models (e.g., user, task, and domain model) rather than operator-interface interaction. To address this issue for better overall human-machine system performance, this section starts with the models an IAI should follow and the needs of an interaction model. A framework for an IAI design and taxonomy to guide the empirical study for optimizing operator-agent interaction is proposed, followed by adaptation and automation considerations from an interface design perspective.

5.1 Models for IAI Design and Implementation

The reality of intelligent interface technology is "indirect management" of information against "direct manipulation" (Benyon & Murry, 2000). Since an IAI is multi-faceted in purpose and nature, the IAI should include capabilities for multimedia input analysis (i.e., mouse, keyboard, spoken language, eye and head tracking, and three-dimensional gesture), multimedia presentation generalization (i.e., windows, menus, dialogue boxes for typed or spoken language, graphics, and gesture), and using user, system and task models to personalize and enhance interactions. To realize the reality of indirect information management and improve the efficiency, effectiveness, and naturalness of human-machine interaction, it is necessary to represent, reason, and act on models of the organization, knowledge (i.e., the user, domain, task, system, and world), agent, communication (e.g., graphic, natural language, gesture), and design.

The models used widely in the knowledge-based systems design techniques (e.g., CommonKADS and EMD) and intelligent interface technology community (Edwards, 2004; Banbury, *et al.* 2007) are as follows.

- Organization model: used in systematic analysis of organizational processes for implementation of a knowledge system. They provide specifications for ontology modeling;
- Knowledge model: abstract representation of knowledge of the context (domain, task, user, system, inference and world);
- Domain model: provides abstract representation of the domain knowledge, relevant to the purpose of the mission;
- Task model: expresses knowledge about tasks being pursued, hierarchy of high-level tasks and sub-tasks;
- User model: expresses the system's knowledge about the user's behavior, knowledge, abilities, needs, and preferences;
- System model: expresses the system' s knowledge about itself and its abilities and the means by which it can assist users;
- World model: expresses the software's knowledge about the external world, i.e., the objects that exist in the world, their properties and the rules that govern them;
- Agent model: incorporates knowledge relating to the participants of the system (i.e., computer and human agents), as well as their roles (Zhu & Zhou, 2006, 2008);
- Communication model: incorporates knowledge of the manner in which communication takes place between the human operator and the system, and between the system agents themselves; and
- Design model: comprises the hardware and software requirements related to the construction of the intelligent adaptive system.

Although the above models consider most aspects of IAI design and implementation requirements, the interaction between operator and agents has not been actively explored. Only after Benyon and Murray (2000) defined the interaction model is the importance of interaction between operator and agents emphasized. This represents the strategies and theory of operator-agent interaction in a particular IAI system in which AIAs are embedded.

"An interaction model is defined a set of principles, rules and properties that guide the design of an interface. It describes how to combine interaction techniques in a meaningful and consistent way and defines the 'look and feel' of the interaction from the user's perspective. Properties of the interaction model can be used to evaluate specific interaction designs" (Beaudouin-Lafon, 2000).

5.2 The Role of Interaction Model

Since the above IAI models explicitly represent the domain of applications, task constraints, and the flexibility inherent in human interaction with a complex system, the interaction models need to reflect the work environment and its dynamic nature, as perceived by the operator given the current system state and goals. Thus, an interaction model should represent at least three properties of both the control and controlled systems as well as the operator: (1) what changes to the system the operator wants to make; (2) why the changes should be made with respect to system goals and current state, and (3) how the needed changes to the system can be made (i.e., the operator activities undertaken to achieve the desired state). In addition, if there are concurrent activities, the model should represent their nature and choices available to the operator given the current system state. To be useful to the design, an effective interaction model must be both descriptive and prescriptive to describe what an operator actually does and to specify what an operator should do next (decision-aid).

Since an IAI contains many AIAs to understand operators, act for operators, and explain and specify adaptations to operators, operator-interface interaction is essentially operator-agent interaction. With the assistances from various automation agents, operators' tasks can be diverted from direct manipulation to indirect information management. Operators' focus will be on overall operation and performance of the system, instead of the detailed functions of the system that should be delegated to various embedded automation agents, i.e., AIAs. These AIAs in the interface should be able to tell operators what is happening (descriptive) and what should or will be done next (prescriptive). They should also be able to learn operators' intentions, monitor their cognitive workload and performance, and guard their resources and time (intelligently). They should also be able to learn from past experiences and change how they behave in any given situation (adaptive). In addition, these AIAs should enable themselves to communicate and cooperate with each other and act according to the results of communications (cooperative). Therefore, with the interaction facilitation, the interface is really an entity that gathers all the AIAs that are not only descriptive/prescriptive, but also adaptive and cooperative, as illustrated in Figure 2.

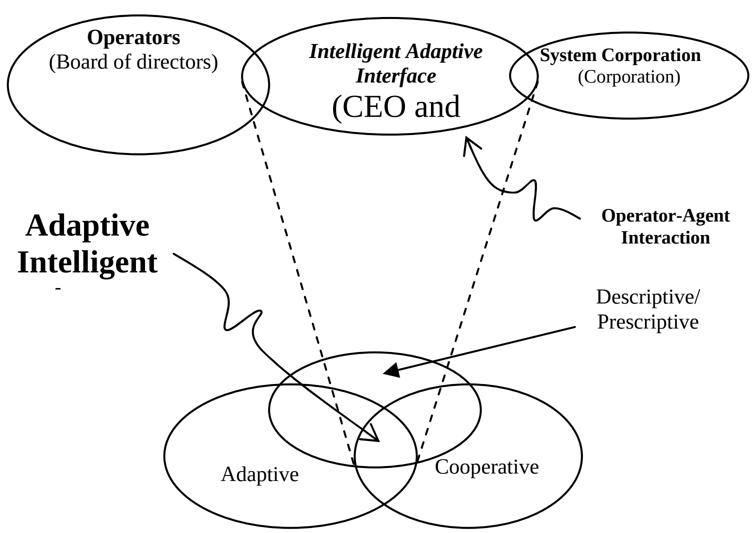

Figure 2. Operator-Agent Interaction model

Comparing the way operators interact with interface agents in a human-machine system with the way a management team runs a corporation, it is apparent that they have a similar fashion of interaction. As shown in Figure2, operators can be regarded as a board of directors of a corporation with the purpose of making profits (maximize overall system performance). The board must trust its management team (components of an interface) and delegate them to run the business. Through the management team (a group of agents), the interface will then act as a bridge between operators (the board) and the system (the corporation) to conduct the mission (running the business) and represent the interest of the operators. Thus, an IAI can be regarded as an entity or communication channel that consists of a group of assistants (AIAs) including a chief executive office and his/her assistants. Through the communication channel (IAI), the management team (AIAs) automatically conducts routine business and keeps the board (operators) updated on what is happening and what the corporation (system) should or will do the next. The team will handle any issues between itself and the system including correcting any mistakes without any interference from the board, except for some emergencies that they have to draw to the board's attention. When that happens, the board must understand what the team is doing (through interactive communications), as well as why something happened, how to be directly involved in the business (take over the system control) and/or instruct the management team to handle situations. In order to do so, the team should have the ability to adapt not only to changes of business environment (system/environment changes), but also to changes of the board's directions/intentions. At the same time, the team should be able to communicate effectively and work collaboratively to increase profits for the corporation (maximizing overall human-machine system performance). Therefore, the interface (management team) has to be the assembly of various AIAs who are descriptive/prescriptive, adaptive, and cooperative.

To carry out the above functions, it should also include a combination of algorithmic and symbolic models for assessing and predicting an operator's activities, awareness, intention, resources, and performance. It should support more rapid task completion by operators with less work (efficiency), provide the right information at the right time and tailor the content and form of the interaction to the context of operators, tasks, dialogue (effectiveness), support spoken, written, and gestural interactions, ideally as if interacting with a human interlocutor (naturalness).

5.3 Optimization of Operator-Agent Interaction

As pointed out by Lajos (1995), more complex adaptation schemes for an IAI should require and/or allow:

- Multiple levels of interaction;
- Multiple dialogues and modes (content and form) of interaction;
- Involvement of many personal shaping factors into adaptation;
- Wide choice of skill, rule and knowledge-based interaction elements; and
- Multiple modularity/hierarchical levels in the interface structure and functions.

Besides visual inputs, other communication channels between operators and interfaces should be considered as well, such as auditory (verbal and aural) inputs. Multimodal inputs (Maat & Pantic, 2007) are good for monitoring and communication between an operator and interface, but the more information being provided and processed, the more stress and workload the operator is under. The appropriate level and channel of input and interaction should depend on the task itself and the current state of interaction. The reliability and accuracy of those operator/user models and algorithms are also critical for system design and interaction optimization. In addition, the optimization should also consider trust and transparency issues. With trust, an agent does things automatically. A dynamic adaptive interface/agent (Scallen, et al. 1995) that has cognitive inference aiding systems should automatically provide information without the requirement for control input by the operator. With transparency, some agents (e.g., some dialogue windows) effectively disappear when necessary, thus, enabling the operator to interact directly with the objects of interest in the domain and to achieve effective interaction with minimal cognitive effort.

To optimize operator-agent interaction in an IAI-based system, the above theoretical issues have to be addressed in empirical investigations and thus provide solid guidelines on IAI design. In order to evaluate the role of AIAs facilitating operator-agent interaction in an IAI-based system, a taxonomy is proposed for guiding any experimentations, as illustrated in Figure3. The taxonomy includes three domain variables of human-machine interfaces: level of (task) complexity, level of (operator) temporal workload, and level of (interface) intelligence/automation. With the context of UAV control, task complexity is decided by the cognitive nature of a task a UAV operator has at a specific position (e.g., UAV pilot, UAV payload operator, and Tactical Navigator) with low, medium, and high levels. Operator temporal workload is decided by the extent of operator time stress and/or the amount of information processing at low, medium, and high levels. Interface intelligence is decided by the agent technology, and should cover all aspects of human perception, behavior, and cognition. The intelligence is also directly related to automation level – fully manual, semi-automatic, and fully autonomous. Since automation changes the level of operators' involvement, how will AIAs' assistances impact an operator's perception, cognition, behavior, and goals when s/he interacts with different tasks at different levels of automation and task complexity? The question of the role of UAV operators playing (task complexity), the level of time stress and/or amount of information (operator workload) or the level of automation (interface intelligence) that will affect overall performance is really the driving force to the design guidelines of IAIs.

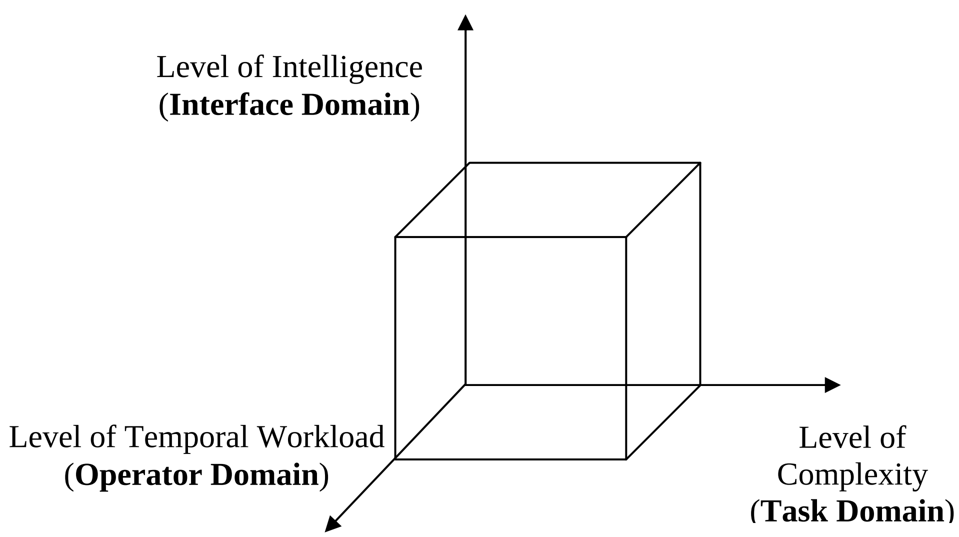

Figure 3. Taxonomy of Operator-Agent Interaction in an AIA-based System.

5.4 A Generic Framework

Considering all the aspects discussed above on the design of an IAI, a generic framework of an AIA-based system can be proposed. As shown in Figure 4, the agent technology plays a crucial role for operator-agent interaction in this hierarchical architecture. Basically, an IAI structure consists of three layers that represent a hierarchy of various AIAs: senior, working, and junior. Each layer has a number of agents working at different levels. The first layer has four senior agents to manage the higher-level tasks for achieving higher-level goals of the overall system (a management team running a corporation). It includes Managing, Interaction, Modeling, and Sensing agents who play the role of managing and coordinating information flow, communications, and feedback among the system, interface, agents, and operators.

The second layer consists of working agents (working level employees of an organization) who work on individual aspect of major tasks for the senior agents to achieve individual goals for tasks at different levels. There are four working agents outlined in Figure 4: Inference (working for Modeling agent), Behavior, Perception, and Cognition (working for Interaction agent). Behavior expresses the manual control of input devices; perception includes the aspects of visual, verbal, aural, tactile, and force; and cognition includes workload, situational awareness, complacency, skill degradation, fatigue, and frustration.

There should be more working agents in the IAI structure (e.g., a fuel monitoring agent working for Sensing agent with UAV control context), but they are not all illustrated in Figure 4 as only the most relevant agents are discussed here.

For the same reason, no junior agent is explicitly outlined in Figure 4. Junior agents (assistants to the working level employees in an organization) are the third layer of IAI hierarchy that works on the details of individual tasks for the working agents. For example, an eye-gazing tracking agent is a junior agent that monitors operator’s eye movements and communicates with its working Perception Agent to deliver eye-gaze tracking information. The details of individual junior agents are out of the scope of this paper.

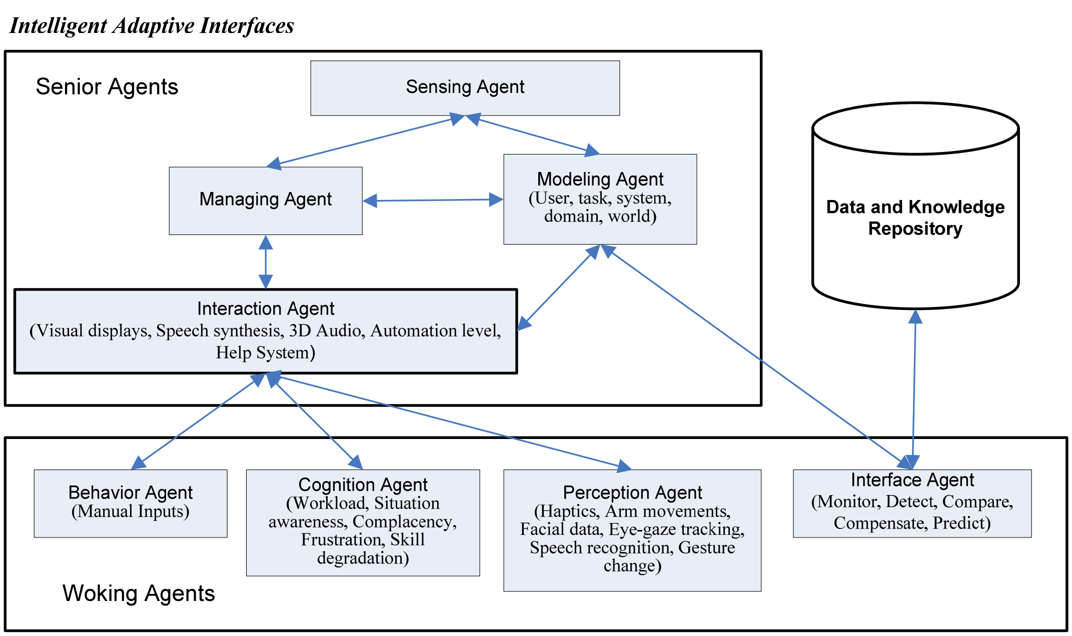

Figure 4. A Framework of Intelligent Agent Interface (the bi-directional arrows represent the communications and feedback).

The functions of different agents in Figure 4 are:

- Senior Agents:
    - Managing Agent: managing all information flow, controlling the display, deciding automation level for agents, coordinating with sensing, modeling, and interaction agents based on the states (knowledge) of all agents (external and internal) and the system itself including error monitoring and emergency control.
    - Sensing Agent: gathering information from internal and external tasks, sensors, data link to keep managing agent and modeling agent updated on current states of internal and external sensors and environments, etc.
    - Modeling Agent: gathering information from other three senior agents about current states of operators, system, and environment. It also provides updated models to various agents according to their own algorithms. Modeling Agent has a working agent – Inference Agent that basically collects information from Sensing, Modeling, Behavior, Perception, and Cognition Agents about the operator, tasks, domain, system, and environment for the comparison with the knowledge in the database – Data and Knowledge Repository.
    - Interaction agent: handles information from interactions with operator and communicates with the managing agent to control the display, automation levels for different agents, and the emergency system (with feedback). It includes three working level agents: behavior, perception, and cognition.
- Working Agents:
    - Behavior Agent focuses on data inputs: keyboard and mouse or any other input device if present. The data will provide information to the cognition agent and communicate with the interaction agent about the cognitive state of the operator through the process of model tracing carried out by the Inference Agent.
    - Perception Agent focuses on low arm movements, facial data processing, eye-gaze tracking, and gestural information. Through the analysis of gathered data, the operator's attention, fatigue, frustration, and even fear or excitement can be interpreted and transferred to the cognition agent for further analysis through the Interaction Agent.
    - Cognition Agent focuses on the analysis of operator's workload, situation awareness, complacency and skill degradation (performance) based on the comparison of embedded operator's models with the information gathered from behavior and perception agents.
    - Inference Agent basically collects information from the Sensing Agent, Modeling Agent, Behavior Agent, Perception Agent, and Cognition Agent about the operator, tasks, domain, system, and environment for the comparison with the knowledge in the database – Data and Knowledge Repository.

It is noted that the bi-directional arrows in Figure 4 represent communication and feedback between agents themselves, interface and operators, and interface and external systems (through Sensing Agent).

In the IAI hierarchy, the working-level agents take major responsibilities for detailed work, and they not only work independently themselves, but also cooperate with other AIAs under the supervision of a senior AIA. For instance, Behavior, Perception, and Cognition agents work independently to collect data about the operator and send them to the Interaction agent. They can also communicate with each other through the Interaction agent for coordination purposes to reflect current states of the operator (description). Other descriptive information is from the Sensing agent about current system states, tasks, and goals. Based on the information described by these two senior agents, the Modeling agent will reason and model the operator, tasks, system, and environment through its working agent – Inference agent. The Inference agent will then classify, compare, and evaluate all the data according to the data in the system database, i.e., the Data and Knowledge Repository in Figure 4. The results will be sent back to the Modeling agent to update and predict possible changes. The Modeling agent will then relay the information to the other three senior agents to facilitate the reasoning process for changes that need to be made. Thus, according to the algorithms and criteria built into these agents and systems, Managing and Interaction senior agents can decide how to make the necessary adjustments (adaptation) when the operators' goals or the environmental situation change (prescription). The changes include decisions on the adaptation method, automation levels, and actions to be taken.

Therefore, the IAI structure can facilitate operator-agent interaction by perceiving, reasoning, interpreting, and predicting the current and future states of the operator, system, and environment. The prescriptive aiding system can also assist decision-making in function allocation and automation levels accordingly. This provides the foundation for AIAs to take some tasks from an operator automatically.

As shown in Figure 4, an AIA can be regarded as a standard computer subroutine with arguments and return values. A Managing agent can be regarded as a main function of a computer program. It has arguments (inputs) from three other senior agents: Modeling, Sensing, and Interaction agents that should have knowledge of the system (interface) itself, task processes, task environments, operator's physical, emotional, and cognitive states, current automation levels of all agents, and emergency situations. Its return (output) will be adjustments to other actions accordingly including the new models chosen by the Modeling agent to adopt for operators, tasks, systems, and interactions. For the Sensing agent, the inputs include all the data from internal and external sensors and data links. The output will be the current and next states of the system and task environments. For the Interaction agent, the inputs will be the states of operator's behavior, perception, and cognition, as well as verbal communication. The output will be possible next stages of the process and recommended action the user and the interface should take. For the Modeling agent, the inputs will be the information from Sensing and Interaction senior agents and its Inference agent. The output will be new models of operator, tasks, system, and environment for the three other senior agents and will be sent back to the database through the Inference agent.

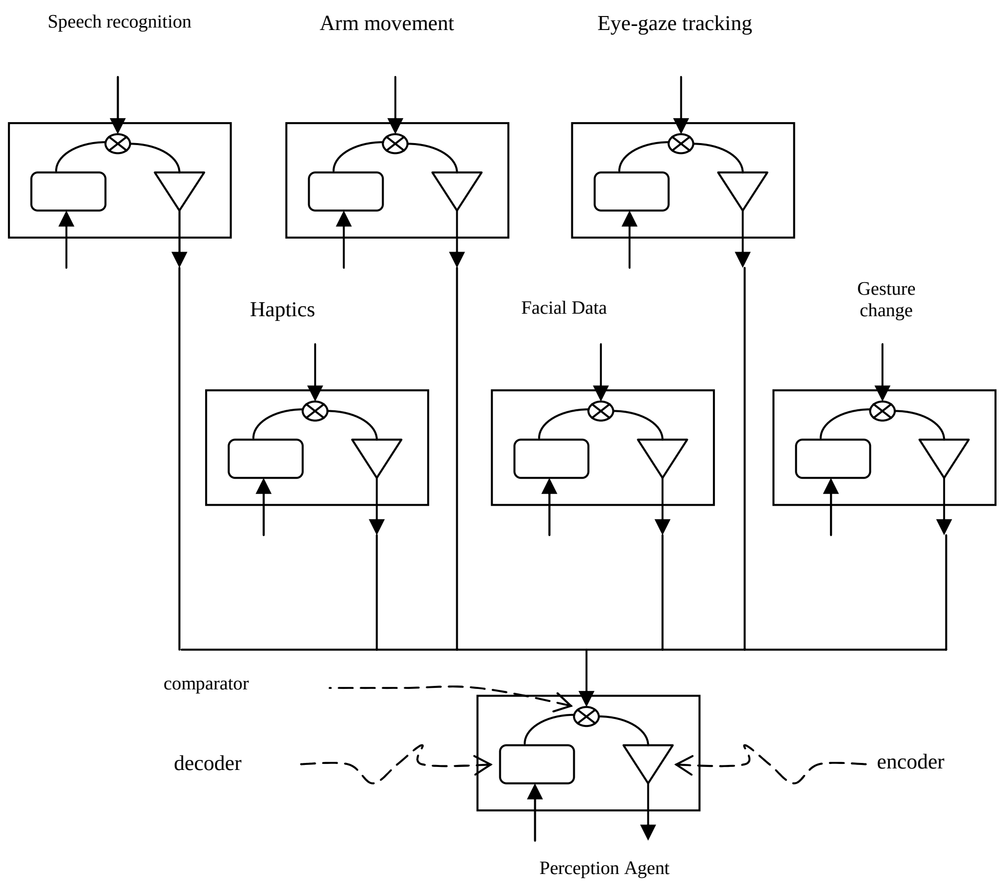

Figure 5. **Computer routine as a standard AIA ("⊗" means a comparator, "▭" means an decoder, and "▽" means an encoder).**

Similarly, the inputs for behavior and perception junior agents include the output from sensors tracking input devices, arm movements, facial data, eye-gaze, and gesture changes, respectively. By working with embedded models, the outputs will be the levels of workload, situation awareness, frustration, fear, excitement, etc. And these outputs will become inputs to the Cognition Agent who will infer further to next stages of workload, situation awareness, complacency, and skill degradation according to cognitive models as well as recommended adjustments the operator or the agent has to make. It is noted that some of these models (e.g., facial expression and speech recognition models) do not work in all environments. Since they can work only for specific users but not generic, there should be some adjustments when applying them in the interface design process.

Figure 5 illustrates an example of an AIA (perception agent) as a standard computer routine with standard arguments and return values. Here, the arguments (inputs) are facial data, states of haptics, arm movement, speech recognition, eye-gaze tracking, and gesture change. Each junior agent has its own input from the sensor, and compares new data with the previous state after decoding based on the embedded models. It then delivers return values (output) to the Perception agent after encoding. The Perception agent gathers all outputs (knowledge) of these junior agents and compares them with its own embodied models. The outputs (cognitive states) will then be sent to the Interaction and Cognition Agents.

5.5 Adaptation Processes

With AIAs' help, the interface should be able to gather knowledge of the operator, system itself, environment, and the process of various tasks. The interface should also be aware of activities, resources, and the operator's intention (Tomlison, et al. 2007), plan, performance, and awareness. In order to have the knowledge and thus build adaptation into the interface, the relevant agents should interact with different aspects of the operator's behaviors (refer to Table II): physiological attributes (eye, ears, fingers, etc.), intellectual characteristics (capacity, recognition, learning, reasoning, decision, trust, etc.), knowledge basis (knowing the environment, system, operator, task, etc.) and psychological states (concentration, situational awareness, vigilance, fatigue, patience, etc.).

From management's perspective, the interface agents (e.g., managing agent) should have the control to allocate appropriate tasks to the operator and automation agents. The function allocation should also follow certain philosophies of adaptation in terms of answering W5+ (i.e., what, who, where, why, when, and how) questions addressed in Section 3.2. A framework for the function allocation/adaptation can then be proposed and illustrated in Figure 6. It includes four major processes: knowledge acquisition, attention, reasoning, and decision-making.

- Knowledge Acquisition: gathers information from the Sensing and Interaction agents about states of the environment, tasks, system, and operators. The purpose is to tell the system and operators what is currently happening.

- Attention: involves the work of two senior agents (Managing and Modeling) and one working agent (Inference). They indicate what changes (adaptation) should or will happen, and where and when they will happen. The purpose is to enable operators to be aware of what should or will be happening in the next stage of the system.

- Reasoning: involves all four senior agents (Managing, Sensing, Interaction, and Modeling) and four working agents (Behavior, Perception, Cognition, and Inference) because the interface needs to gather operators' or system's information to understand and interpret why all changes are necessary. They provide information on operators' and system's states and models of the operator, task, environment, and system. The information is obtained through monitoring the states of operators, sensors, tasks, environment, and system. The Inference agent compares current states of these variables with existing models in the database, judges whether changes are necessary, and suggests what models should be used for adaptation through the Modeling agent with predicted outputs.

- Decision-Making: involves both Interaction and Managing senior agents to evaluate proposed methods of changes from the reasoning process by applying two criteria. The primary criterion looks at human performance consequences: mental workload, situation awareness, complacency, and skill degradation. The secondary criterion includes the automation reliability and costs of consequences (Parasuraman, et al. 2000). Once confirmed with the criteria, the Decision-Making process will choose a method of adaptation and identify automation levels for various agents. The level of automation will be discussed in detail in the next section.

The primary goal of facilitating these processes is to optimize operator-agent interaction through adaptation by providing the operator's and system's states including: activities, awareness, intentions, resources, performance, and time.

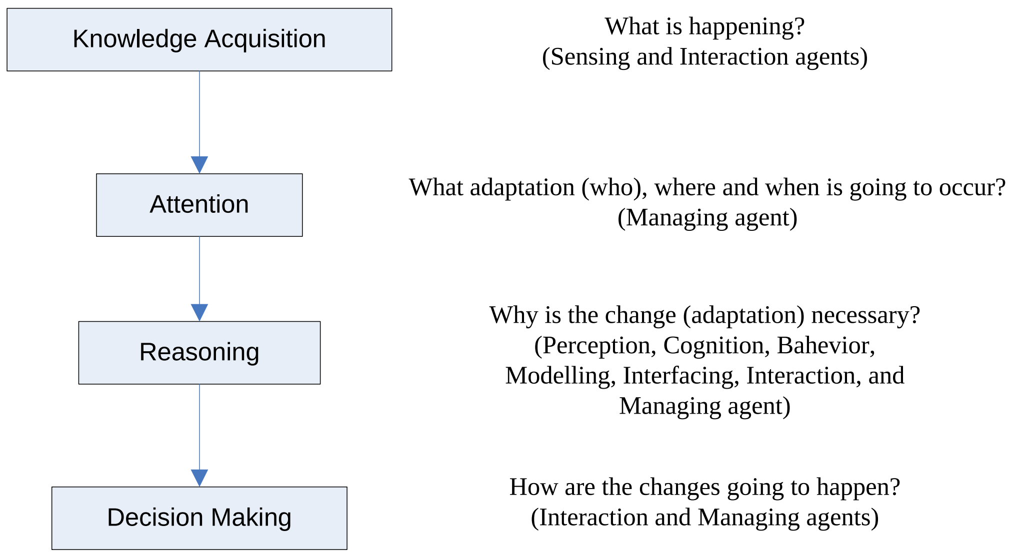

Figure 6. Adaptation and Automation Framework.

6. Conclusions

Network Enabled Operation (NEO) is a fundamental paradigm shift from platform-centric operations (PCO). NEO is defined as an information superiority-enabled concept of operations that generates increased combat power by networking sensors, decision-makers, and operators. This entails shared situation awareness, increased speed of command, higher tempo of operations, increased survivability, and a degree of self-synchronization. In essence, NEO translates information superiority into operational power by effectively linking knowledgeable entities in the battle space. Consequently, as the number of entities, or systems, in the battle space increases, the resulting NEO system-of-systems will likely exhibit more complex emergent behaviour which needs to be properly addressed as we move from PCO to NEO.

Software agent technology and the definition of agent are examined. Adaptive intelligent agent (AIA) and intelligent adaptive interface (IAI) are then properly defined in the context of human-machine interaction to avoid any confusion. The differences between human-human interaction, human-machine interaction, and operator-agent interaction are also reviewed to understand the nature of human-computer interaction as operator-agent interaction for the purpose of maximizing the overall human-machine system performance.

To address the challenges, a framework of intelligent adaptive interface was proposed to optimize operator-agent interaction based on the philosophy of proactive use of automation and user-oriented design principles. The active use of AIAs is to take on some responsibilities from users who still have appropriate amount of control tasks to maintain situational awareness. Thus, both users and the IAI can work actively and collaboratively as partners to achieve maximum overall performance. The objective for designing such a system is to provide users with a *descriptive and prescriptive* decision-aid by revealing what the system or users are actually doing and specifying what the users or the system should do next.

One concept being pursued by the Canadian Forces is to introduce Uninhabited Air Vehicles (UAVs) as a new Integrated Intelligence, Surveillance, and Reconnaissance (IISR) platform. Their control involves a variety of workload intensive activities that potentially impose severe constraints and excessive workload on operators. To address this issue, suitable supporting technologies combining operators and automation need to be investigated to satisfy mission requirements (real-time, workload, manning reduction, etc.). IAIs for decision support are sufficiently mature technologies that increase human information processing capabilities in all of the most critical applications from piloting displays and sensor operators to those involved in IISR activities. However, the design of IAIs is extremely challenging as there are no established design guidelines for this type of advanced interfaces. This document looked at currently available theories and the requirements for designing an IAI, and proposed a generic framework for guiding the interface design and further verification of its effectiveness.

First, software agent technology and the definition of agent were examined. AIA and IAI are then defined in the context of human-machine interaction to avoid any confusion. The differences between human-human, human-machine, and operator-agent interactions are also reviewed to understand the nature of human-machine interaction as operator-agent interaction for the purpose to maximize overall human-machine system performance.

Through the review of agent-based interface design approaches and two empirical investigations on IAI technology, CommonKADS and EMD are recommended as the baseline interface design frameworks along with IDEF and PCT for analyzing the models of the organization, system, domain, tasks, goals, and various structures of AIAs for the purpose of implementing an IAI. CTA and EID approaches can also be used for the specifications of the interface at the cognitive control levels for a safe and reliable system. The challenges of IAI design are also recognized from different aspects: usability principles, development methods, design practice, and scalability.

To address the challenges, a multiple-agent system framework is proposed to optimize operator-agent interaction based on the philosophy of the proactive use of automation and user-centered design principles. The active use of AIAs is to take some responsibilities from users who are still in the control to maintain situation awareness. Thus both users and IAI can work actively and collaboratively as partners to achieve maximum overall performance. The objective for designing such a system is to provide users a descriptive and prescriptive decision aid by telling what the system or users are actually doing and specifying what users or the system should do next.

To achieve the goal of helping users increase situation awareness and making right decisions, the proposed hierarchal framework has three levels of AIAs: senior, working, and junior agents that monitor user needs and system status. The four senior agents oversee the internal and external system and environment, manage the information flow and operation routines, and communicate with internal and external sensors, operators, and other embedded automation agents. The working agents gather information from both the senior and junior level agents, then reason and model the states of operators, tasks, system, and environment in order to provide feedback and decision-aids to the senior level agents when necessary. The junior level agents act on the direct inputs from the operator, system, and environment to provide information to the working level agents for further analysis and decision-aids.

To facilitate the optimization of operator-agent interaction, the communication between the two has to be as natural as possible, i.e., resembling human-human interaction. To keep an effective and efficient partnership when AIAs actively take some tasks, an operator should not be a passive observer but an active controller being involved in the adaptation processes of knowledge acquisition, attention, reasoning, and decision-making.

Bibliography

[1] Abascal, J., Castro, I. F. d., Lafuente, A., & Cia, J. M. (2008). Adaptive Interfaces for Supportive Ambient Intelligence Environments In Computers Helping People with Special Needs (Vol. 5105, pp. 30-37). Berlin, Germany: Springer.

[2] Arai, F., Fukuda, T., Yamamoto, Y., Naito, T., & Matsui, T. (1993). Interactive adaptive interface using recursive fuzzy reasoning, IEEE Annual International Symposium on Virtual Reality (pp. 104-110). Seattle, WA, USA.

[3] Bainbridge, L. (1983). Ironies of Automation. Automatica, 19(6), 775-779.

[4] Banbury, S., Gauthier, M., Scipione, A., & Hou, M. (2007). Intelligent Adaptive Systems: Literature Research of Design Guidance for Intelligent Adaptive Automation and Interfaces (No. A112084): Defence Research and Development Toronto, Canada.

[5] Bar-Yam, Y. (1997). Dynamics of Complex Systems, West View Press, 1997.

[6] Bar-Yam, Y.(2003). Complexity of Military Conflict: Multiscale Complex Systems Analysis of Littoral Warfare", April, 2003.

[7] Bar-Yam, Y. (2004). Making Things Work: Solving Complex Problems in a Complex World, NECSI Knowledge Press, 2004.

[8] Bates, J. (1994). The role of emotion in believable agents. Communications of the ACM, 37(7), 122-125.

[9] Baylor, A. L., Rosenberg-Kima, R. B., & Plant, E. A. (2006). Interface agents as social models: the impact of appearance on females' attitude toward engineering, Conference on Human Factors in Computing Systems (pp. 526 - 531). Montréal, Québec, Canada ACM New York, NY, USA.

[10] Beaudouin-Lafon, M. (2000). Instrumental interaction: an interaction model for designing post-WIMP user interfaces, The SIGCHI conference on Human factors in computing systems (pp. 446 - 453). The Hague, The Netherlands.

[11] Bennett, K. B., & al, e. (2001). A theoretical analysis and preliminary investigation of dynamically adaptive interfaces. The International Journal of Aviation Psychology, 11(2), 169-195.

[12] Bennett, K. B., & Hettinger, S. (2001). A theoretical analysis and preliminary investigation of dynamically adaptive interfaces. The International Journal of Aviation Psychology, 11(2), 169-195.

[13] Benyon, D. R., & Murray, D. M. (1988). Experience with adaptive interfaces. The Computer Journal, 31(5), 465-473.

[14] Benyon, D. R., & Murray, D. M. (1993a). Adaptive systems: From intelligent tutoring to autonomous agents. Knowledge-Based Systems, 6(4), 197-219.

[15] Benyon, D. R., & Murray, D. M. (1993b). Applying user modeling to human-computer interaction design. Artificial Intelligence Review, 7(0), 199-225.

[16] Benyon, D. R., & Murray, D. M. (2000). Special issue on intelligent interface technology: editor's introduction. Interacting with Computers, 12(4), 315-322.

[17] Berkan, R. C., Upadhyaya, B. R., Tsoukalas, L. H., & Kisner, R. A. (1991). Advanced automation concepts for large-scale systems. IEEE Control Systems Magazine, 11(6), 4-12.

[18] Bonneh, Y. S., Cooperman, A., & Sagi, D. (2001). Motion-induced blindness in normal observers. Nature, 411, 798-801.

[19] Bernon, C., Cossentino, M., Gleizes, M.-P., Turci, P., & Zambonelli, F. (2005). A Study of Some Multi-agent Meta-models In Agent-Oriented Software Engineering V (Vol. 3382, pp. 62-77). Berlin, Germany: Springer.

[20] Bosse, T., Lam, D. N., & Barber, K. S. (2008). Tools for analyzing intelligent agent systems. Web Intelligence and Agent Systems: An International Journal, 6, 355–371.

[21] Bowan, D. A., & Hodges, L. F. (1999). Formalizing the Design, Evaluation, and Application of Interaction Techniques for Immersive Virtual Environments. Journal of Visual Languages and Computing, 10(1), 7.

[22] Bradshaw, J. (1997). Software agents. Cambridge, MA: MIT Press.

[23] Bratman, M. E., Isreal, D. J., & Pollack, M. E. (1988). Plans and resource-bounded practical reasoning. Computational Intelligence, 4(0), 349-355.

[24] Browne, D., Totterdell, P., & Norman, M. (1990). Adaptive User Interfaces. London, UK: Academic Press.

[25] Burgos, D., Tattersall, C., & Koper, E. J. R. (2007). Representing adaptive and adaptable Units of Learning. How to model personalized eLearning in IMS Learning Design. In B. F. Manjon, J. M. S. Perez, J. A. G. Pulido, M. A. V. Rodriguez & J. Bravo (Eds.), Computers and Education: E-learning - from theory to practice (pp. Avail: http://dspace.ou.nl/dspace/bitstream/1820/1785/1821/BURGOSetAl_BookChapter_v1823_1821August1806_ToSend.pdf). Germany: Kluwer.

[26] Burns, C. M., & Hajdukiewicz, J. R. (2004). Ecological Interface Design. Boca Raton, FL: CRC Press.

[27] Byrne, E. A., & Parasuraman, R. (1996). Psychophysiology and adaptive automation. Biological Psychology, 42(0), 249-268.

[28] Carolis, B. D., Cozzolongo, G., Pizzutilo, S., & Plantamura, V. L. (2005). Agent-Based Home Simulation and Control In Foundations of Intelligent Systems (Vol. 3488, pp. 404-412). Berlin, Germany: Springer.

[29] Chu, Y., & Rouse, W. (1979). Adaptive allocation of decision making responsibility between human and computer in multitask situations. IEEE Transactions on Systems, Man, and Cybernetics, 9(12), 769-778.

[30] Cohen, P. R., & Levesque, H. J. (1990). Intention is choice with commitment. Artificial Intelligence, 42(0), 213-261.

[31] Cooper, M., & Hockley, A. (1986). The evaluation of an adaptive interface. In Iee (Ed.), Colloquium on Adaptive Man-Machine Interfaces (pp. 4/1-2). London, UK: IEE.

[32] Dam, S. B. J. V., Mulder, M., & Paassen, M. M. V. (2008). Ecological Interface Design of a Tactical Airborne Separation Assistance Tool. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 38(6), 1221-1233.

[33] Dimitrova, M., & al, e. (2001). A framework for integration of knowledge and style support in user-adaptive interface. In G. Johannsen (Ed.), IFAC Analysis, Design and Evaluation of Human-Machine Systems (pp. 525-529). Kassel, Germany: Pergamon.

[34] Duric, Z., Gray, W. D., Heishman, R., Li, F., Rosenfeld, A., Schoelles, M. J., et al. (2002). Integrating perceptual and cognitive modeling for adaptive and intelligent human-computer interaction. Proceedings of IEEE, 90(7), 1272-1289.

[35] Edwards, J. L. (2004). A Generic, Agent-Based Framework for the Design and Development of UAV/UCAV Control Systems: DRDC Toronto.

[36] Edwards, J. L., & Hendy, K. C. (2000). A testbed for intelligent aiding in adaptive interfaces from https://www.aaai.org/Papers/Symposia/Spring/2000/SS-00-01/SS00-01-008.pdf.

[37] Eklund, J., & Sinclair, K. (2000). An empirical appraisal of the effectiveness of adaptive interfaces for instructional systems. Educational Technology and Society, 3(4), 165-177.
[38] Finin, T., Nicholas, C., & Mayfield, J. (1997). Agent-based information retrieval, The 2nd ACM Conference on Digital Libraries (pp.???). Philadelphia, PA.
[39] Fukuda, T., & al, e. (1992). Concept and realization of adaptive interfaces based on user's skill and uncomfortableness, IEEE International Workshop on Robot and Human Communication (pp. 171-176). New York, NY.
[40] Gajos, K. Z., Czerwinski, M., Tan, D. S., & Weld2, D. S. (2006). Exploring the design space for adaptive graphical user interfaces, The working conference on advanced visual interfaces (pp. 201 - 208). Venezia, Italy
[41] Galster, S. M., & Bolia, R. S. (2002). Effects of information automation and decision-aiding cueing on action implementation in a visual search task, the Human Factors and Ergonomics Society 46th Annual Meeting (pp. pp 438-442). Santa Monica, CA.
[42] Geneseveth, M., & Ketchpel, S. (1994). Software agents. Communications of the ACM, 37(7), 48-53.
[43] Georgeff, M. R., & Lansky, A. L. (1987). Reactive reasoning and planning, The 6th National Conference on Artificial Intelligence (AAAI-87) (pp. 677-682). Seattle, WA.
[44] Georgeff, M. R., Pell, B., Pollack, M., Tambe, M., & Wooldridge, M. (2000). The belief-desire-intention model of agency. In Intelligent Agents V: Agents Theories, Architectures, and Languages (Vol. 1555, pp. 1-10). Berlin, Germany: Springer.
[45] Gibson, J. J. (1979). The Ecological Approach to Visual Perception. Boston, MA: Houghton Mifflin.
[46] Gil, A. E., Passino, K. M., & Ganapathy, S. (2008). Cooperative Task Scheduling for Networked Uninhabited Air Vehicles. IEEE Transactions on Aerospace and Electronic Systems, 44(2), 561-581.
[47] Glass, A., McGuinness, D. L., & Wolverton, M. (2008). Toward establishing trust in adaptive agents, Proceedings of the 13th international conference on Intelligent user interfaces (pp. 227-236). Gran Canaria, Spain.
[48] Gómez-Sanz, J. J., & Garijo, J. P. F. (2000). Intelligent interface agents behaviour modelling, Mexican International Conference on Artificial Intelligence (pp. 598-609). London, UK.
[49] Grossklags, J., & Schmidt, C. (2006). Software agents and market (in) efficiency: a human trader experiment. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews., 36(1), 56-67.
[50] Guyot, P., & Honiden, S. (2006). Agent-Based Partipatory Situations: Merging multi-agent systems and role-playing games Journal of Artificial Societies and Social Simulation, 9(4).
[51] Hammer, J. M. (1989). Integrating a decision aid and an intelligent interface, IEEE International Conference on Systems, Man and Cybernetics (pp. 1290-1295). Cambridge, MA.
[52] Hancock, P. A., & Chignell, M. H. (1985). An adaptive human-machine system, International Conference on Cybernetics and Society (pp. 305-345). Tucson, AR.
[53] Hancock, P. A., & Chignell, M. H. (1988). Mental workload dynamics in adaptive interface design. IEEE Transactions on Systems, Man, and Cybernetics, 18(4), 647-658.
[54] Hancock, P. A. C., & Chignell, M. H. (1993). Adaptive function allocation by intelligent interfaces, Intelligent User Interfaces'93 (pp. 227-229). Orlando, FL.
[55] Hassell, J., & Harrison, M. (1994). Generalisation and the adaptive interface, People and Computers IX: Proceedings of HCI '94 (pp. 227-229). Glasgow, UK.

[56] Haynesa, G.-S. R., Cohenb, M. A., & Rittera, F. E. (2009). Designs for explaining intelligent agents. International Journal of Human-Computer Studies, 67(1), 90-110.
[57] Hill, R., Polovina, S., & Shadija, D. (2006). Transaction Agent Modelling: From Experts to Concepts to Multi-Agent Systems. In Conceptual Structures: Inspiration and Application (Vol. 4068, pp. 247-259). Berlin, Germany: Springer.
[58] Hook, K. (2000). Steps to take before intelligent user interfaces become real. Interacting with Computers, 12(4), 409-426.
[59] Hou, M., Kobierski, R. D., & Brown, M. (2007). Intelligent Adaptive Interfaces for the Control of Multiple UAVs. Journal of Cognitive Engineering and Decision Making, 1(3), 327-362.
[60] Iglesias, C. A., Garijo, M., Gonzalez, J. C., & Velasco, J. R. (1998). Analysis and design of multiagent systems using MAS-CommonKADS. In M. P. Singh, A. Rao & M. J. Wooldridge (Eds.), Intelligent Agents IV: Agent Theories, Architectures, and Languages (pp. 313-328). Berlin, Germany: Springer.
[61] Innocent, P. R. (1982). Towards self-adaptive interface systems. International Journal of Man-Machine Studies, 16(3), 287-299.
[62] Jamieson, G. A., Miller, C. A., Ho, W. H., & Vicente, K. J. (2007). Integrating Task- and Work Domain-Based Work Analyses in Ecological Interface Design: A Process Control Case Study. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 37(6), 887-905.
[63] Jaquero, V. L., Montero, F., Molina, J. P., & González, P. (2009). Intelligent User Interfaces: Past, Present and Future. In C. B. Miguel Redondo, Manuel Ortega (Ed.), Engineering the User Interface: : From Research to Practice (pp. 259-270). London: Springer.
[64] Jarvis, B., Corbett, D., & Jain, L. C. (2005). Beyond Trust: A Belief-Desire-Intention Model of Confidence in an Agent’s Intentions In Knowledge-Based Intelligent Information and Engineering Systems (Vol. 3682, pp. 844–850). Berlin, Germany: Springer.
[65] Jonassen, D. H., Tessmer, M., & Hannum, W. H. (1999). Task Analysis Methods and Instructional Design. Mahwah, NJ: Lawrence Erlbaum Associates.
[66] Jones, P. M., Chu, R. W., & Mitchell, C. M. (1995). A methodology for human-machine systems research: knowledge engineering, modelling and simulation. IEEE Transactions on Systems, Man, and Cybernetics, 25(7), 1025-1038.
[67] Kaber, D. B., Segall, N., Green, R. S., Entzian, K., & Junginger, S. (2006). Using multiple cognitive task analysis methods for supervisory control interface design in high-throughput biological screening processes Cognition, Technology & Work, 8(4), 237-252.
[68] Keinonen, T. (2008). User-Centered Design and Fundamental Need, NordiCHI 2008: (pp. 211-220). Lund, Sweden.
[69] Keirn-Schreckenghost, D., & Thronesbery, C. (2000). Adaptive interfaces to adjustable control autonomy, The AAAI 2000 Spring Symposium on Adaptive User Interfaces (pp. 54-58). Stanford, CA.
[70] Khoshnevis, B., & Austin, W. (1987). An intelligent interface for system dynamics modeling, The Conference on AI and Simulation (pp. 81-86). San Diego, CA.
[71] Lajos, B. (1995a). Adaptive human-computer interfaces for man-machine interaction in computer-integrated systems. Computer Integrated Manufacturing Systems, 8(2), 133-142.
[72] Lajos, B. (1995b). Adaptive interfaces for human-computer interaction: a colorful spectrum of present and future options, Intelligent Systems for the 21st Century: 1995 IEEE

International Conference on Systems, Man, and Cybernetics (pp. 292-297). Vancouver, BC, Canada.
[73] Lamberti, D. M., & Wallace, W. A. (1990). Intelligent interface design: An empirical assessment of knowledge presentation in expert systems. MIS Quarterly, 14(3), 279-311.
[74] Langley, P. (1999). User modelling in adaptive interfaces, The 7th International Conference on User Modelling (pp. 357-370). Banff, Alberta.
[75] Laurel, B. (1990). Interface agents. In B. Laurel (Ed.), The Art of Human-Computer Interface Design (pp. 355-365). Workingham, UK: Addison-Wesley.
[76] Letsu-Dake, E., & Ntuen, C. A. (2009). A conceptual model for designing adaptive human-computer interfaces using the living systems theory. Systems Research and Behavioral Science, 26(1), 15 - 27.
[77] Lieberman, H. (2002). Interfaces that give and take advice. In J. M. Carroll (Ed.), Human-Computer Interaction in the New Millennium (pp. 475-485). New York, NY: ACM Press.
[78] Lipa, J. F. (1999). Application of advanced displays and intelligent interfaces technology. Rome, NY: AFRL/IFSA.
[79] Liu, J., Wong, C. K., & Hui, K. K. (2003). An adaptive user interface based on personalized learning. IEEE Intelligent Systems, 18(2), 52-57.
[80] López-Jaquero, V., Vanderdonckt, J., Montero, F., & González, P. (2008). Towards an Extended Model of User Interface Adaptation: The Isatine Framework In Engineering Interactive Systems (Vol. 4940, pp. 374-392). Berlin, Germany: Springer.
[81] Lydiard, T. J. (1995). Using IDEF3 to capture the air campaign planning process, from http://www.aiai.ed.ac.uk/~arpi/ACP-MODELS/ACP-OVERVIEW/96-MAR/HTML-DOC/idef3.ps.
[82] Maat, L., & Pantic, M. (2007). Gaze-X: Adaptive, Affective, Multimodal Interface for Single-User Office Scenarios. In Artificial Intelligence for Human Computing, LNCS (Vol. 4451, pp. 251-271). Berlin, Germany: Springer.
[83] Malone, T., Lai, K., & Fry, C. (1995). Experiments with OVAL: A radically tailorable tool for cooperative work. ACM Transactions on Information Systems, 13(2), 177-205.
[84] Manning, S. D., Rash, C. E., LeDuc, P. A., Noback, R. K., & McKeon, J. (2004). The Roles of Human Causal Factors in U.S. Army Unmanned Aerial Vehicle Accidents. Fort Detrick, MD, USA: U.S. Army Aeromedical Research Laboratory and U.S. Army Safety Center.
[85] Maybury, M. (1999). Intelligent User Interfaces: An Introduction, Int'l Conf. on Intelligent User Interfaces (pp. 3-4). Redondo Beach, Los Angeles, CA.
[86] Mayer, R. J., Menzel, C. P., Painter, M. K., deWitte, P. S., Blinn, T., & Perakath, B. (1995). Information integration for concurrent engineering (IICE) IDEF3 process description capture method report. College Station, TX: Knowledge Based Systems Inc.
[87] McGrenere, J., Baecker, R. M., & Booth, K. S. (2002). An Evaluation of a Multiple Interface Design Solution for Bloated Software, ACM Conference on Human Factors in Computing Systems (pp. 164 - 170). Minneapolis, Minnesota, USA.
[88] McNeese, M. D., & Zaff, B. S. (1991). Knowledge as design: a methodology for overcoming knowledge acquisition bottlenecks in intelligent interface design, Human Factors Society 35th annual meeting (pp. 1181-1185). Amsterdam, The Netherlands.
[89] McTear, M. F. (1993). User modelling for adaptive computer systems: a survey of recent developments. Artificial Intelligence Review, 7(0), 157-184.
[90] McTear, M. F. (2000). Intelligent interface technology: from theory to reality? Interacting with Computers, 12(0), 323-336.

[91] Meyer, B., Yakemovic, K. C. B., & Harris, M. (1993). Issues in practical application of an adaptive interface, The 1st international conference on Intelligent user interfaces (pp. 251-254). Orlando, Florida, United States.

[92] Miller, C., & Parasuraman, R. (2007). Designing for Flexible Interaction Between Humans and Automation: Delegation Interfaces for Supervisory Control. Human Factors, 49(1), 57-75.

[93] Miller, C. A. (2005). Delegation Architectures: Playbooks and Policy for Keeping Operators in Charge, The International Conference on Automated Planning & Scheduling. Monterey, CA, USA.

[94] Mitchell, C. M. (1987). GT-MSOCC: a domain for research on human-computer interaction and decision aiding in supervisory control systems. IEEE Transactions on Systems, Man, and Cybernetics, 17(4), 553-572.

[95] Mitchell, C. M. (1996). Models for the design of human interaction with complex dynamic system. In Cognitive Engineering Systems in Process Control (pp. 230-237). Kyoto, Japan.

[96] Mitrovic, N., Royo, J. A., & Mena, E. (2007). Performance Analysis of an Adaptive User Interface System Based on Mobile Agents. In J. Gulliksen, M. B. Harning, P. Palanque, G. C. v. d. Veer & J. Wesson (Eds.), Engineering Interactive Systems (pp. 1-18). Berlin, Germany: Springer.

[97] Moray, N., Inagaki, T., & Itoh, M. (2000). Adaptive automation, trust, and self-confidence in fault management of time-critical tasks. Journal of Experimental Psychology: Applied, 6(1), 44-58.

[98] Mura, N., & Francois, P. (1996). An object-oriented approach for an adaptive interface with dual-classification of the user, The 6th Australian Conference on Computer-Human Interaction (pp. 0279). Hamilton, New Zealand.

[99] Murray, D. M. (1990). Modelling for adaptivity. In M. J. Tauber & D. Ackermann (Eds.), Mental Models and Human-Computer Interaction 2. Amsterdam, Netherlands: Elsevier.

[100] Norcio, A. F. (1991). Adaptive interfaces: modelling tasks and users, IEEE International Conference on Systems, Man and Cybernetics (pp. 1099-1103). Charlottesville, VA.

[101] Norcio, A. F., & Stanley, J. (1989). Adaptive human-computer interfaces: a literature survey and perspective. IEEE Transactions on Systems, Man and Cybernetics, 19(2), 399 - 408

[102] Norman, D. A. (2005). Human-Centered Design Considered Harmful. Interactions, 12(4), 14-19.

[103] Parasuraman, R., & et al. (1992). Theory and design of adaptive automation in aviation systems, Warminster, PA. : Naval Air Warfare Centre - Aircraft Division.

[104] Parasuraman, R., Sheridan, T. B., & Wickens, C. D. (2000). A model for types and levels of human interaction with automation. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 30(3), 286-297.

[105] Powers, W. T. (1973). Behavior: The control of perception. New York, NY: Aldine De Gruyter.

[106] Powers, W. T., Clark, R. K., & McFarland, R. L. (1960). A general feedback theory of human behaviour: Part 1. Perceptual and Motor Skills, 11(0), 71-88.

[107] Richardson, R. J., & Powers, W. T. (1990). Introduction to modern psychology: The control-theory view. Gravel Switch, KY: The Control Systems Group.

[108] Robert R. Hoffman. (1998). Whom (or what) do you (mis)trust?: historical reflections on thepsychology and sociology of information technology, Fourth Symposium on Human Interaction with Complex Systems (pp. 4-14). Dayton, OH, USA.
[109] Rogers, K. (1999). Crash result of pilot losing track of another jet. Las Vegas Review-Journal, p. 3B, .
[110] Rouse, W. B. (1988). Adaptive aiding for human/computer control. Human Factors, 30(4), 431-443.
[111] Rouse, W. B., Geddes, N. D., & Curry, R. E. (1987). An architecture for intelligent interfaces: outline of an approach to supporting operators of complex systems. Human-Computer Interaction, 3(2), 87-122.
[112] Ruff, H. A., Narayana, S., & Draper, M. H. (2002). Human interaction with levels of automation and decision-aid fidelity in the supervisory control of multiple simulated unmanned air vehicles. Presence, 11(4), 335-351.
[113] Saget, S., Legras, F., & Coppin, G. (2008a). Collaborative model of interaction and Unmanned Vehicle Systems' interface, workshop on "Supervisory Control in Critical Systems Management" - 3rd International Conference on Human Centered Processes. Delft, Netherlands.
[114] Saget, S., Legras, F., & Coppin, G. (2008b). Cooperative Interface for a Swarm of UAVs, from http://arxiv.org/PS_cache/arxiv/pdf/0811/0811.0335v1.pdf
[115] Scallen, S. F., Hancock, P. A., & Duley, J. A. (1995). Pilot performance and preference for short cycles of automation in adaptive function and allocation. Applied Ergonomics, 26(6), 397-403.
[116] Schraagen, J. M., Chipman, S. F., & Shalin, V. L. (2000). Cognitive Task Analysis. Mahwah, NJ: Lawrence Erlburm Associates.
[117] Schreiber, G., Akkermans, H., Anjewierden, A., Hoog, R. d., & Shadbolt, N. (2000). Knowledge engineering and management. Boston, MA: The MIT Press.
[118] Schulte, A., Meitinger, C., & Onken, R. (2009). Human factors in the guidance of uninhabited vehicles: oxymoron or tautology? . Cognition, Technology & Work, 11(1), 71-86.
[119] Seong, Y., & Bisantz, A. M. (2002). Judgment and trust in conjunction with automated decision aids: A theoretical model and empirical investigation, The Human Factors and Ergonomics Society 46th Annual Meeting (pp. 593-600). Baltimore, MD.
[120] Shneiderman, B., & Maes, P. (1997). Direct manipulations vs. interface agents. Interactions, 4(6), 42-61.
[121] Siebra, S., & Ramalho, G. L. (1999). ATHENA: A user-centered adaptive interface, International Conf. on Human-Computer Interaction (pp. 346-350). Munich, Germany.
[122] Sternberg, R. J. (1996). Cognitive Psychology. Orlando, FL: Harcourt Brace College Publishers.
[123] Sudeikat, J., Braubach, L., Pokahr, A., Lamersdorf, W., & Renz, W. (2007). On the Validation of Belief–Desire–Intention Agents. In Lecture Notes in Computer Science, Programming Multi-Agent Systems (Vol. 4411, pp. 185-200): Springer Berlin / Heidelberg.
[124] Sukaviriya, P. (1993). From user interface design to the support of intelligent and adaptive interfaces: an overhaul of user interface software infrastructure. Knowledge-Based Systems, 6(4), 220-229.
[125] Sutcliffe, A., Kurniawan, S., & Shin, J.-E. (2006). A Method and Advisor Tool for Multimedia User

[126] Interface Design. International Journal of Human-Computer Studies, 64(4), 375-392
[127] Taylor, R. M. (2001). Cognitive Cockpit systems engineering: Pilot authorisation and control of tasks, from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.12.119&rep=rep1&type=pdf
[128] Taylor, R. M., & Howells, H. (2000). The cognitive cockpit: operational requirement and technical challenge
[129] In S. Robertson, P. T. McCabe & M. Hanson (Eds.), Contemporary ergonomics (pp. 55-60). London, UK: CRC Press LLC.
[130] Tomlinson, G.-B., Baumer, E., Yau, M. L., Alpine, P. M., Canales, L., Correa, A., et al. (2007). Dreaming of adaptive interface agents, ACM Conference on Human Factors in Computing Systems (pp. 2007 - 2012). San Jose, CA, USA ACM.
[131] Tyler, S. W. (1990). Adaptive Interfaces. Palo Alto, CA: Lockheed AI Center.
[132] Tyler, S. W., & True, S. (1986). Adaptive interface design: A symmetric model and a knowledge-based implementation. SIGOIS Bulletin, 7(2-3), 53-60.
[133] Vicente, K. J., & Rasmussen, J. (1992). Ecological interface design: theoretical foundations. IEEE Transactions on Systems, Man, and Cybernetics, 22(4), 589-606.
[134] Weld, S. D. (1995). The role of intelligent systems in the national information infrastructure. Artificial Intelligence Magazine, 16(3), 45-64.
[135] Wickens, C. D., & Hollands, J. G. (1999). Engineering Psychology and Human Performance (3rd Edition). Upper Saddle River, NJ: Prentice Hall.
[136] Wilson, G. F., & Russell, C. A. (2007). Performance Enhancement in an Uninhabited Air Vehicle Task Using Psychophysiologically Determined Adaptive Aiding. Human Factors, 49(6), 1005-1018.
[137] Woodridge, M. (2002). An Introduction to MultiAgent Systems. Chichester, West Sussex, UK: John Wiley & Sons.
[138] Wooldridge, M. J. (1999). Intelligent Agents. Cambridge, MA: MIT Press.
[139] Xiao, Y., Milgram, P., & Doyle, D. J. (1997). Planning Behavior and its Functional Role in Interactions with Complex Systems. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 37(6), 313-324.
[140] Ye, N. and G. Salvendy (1991). An adaptive interface design using neural networks, in Human Aspects in Computing: Design and Use of Interactive Systems and Work with Terminals, H.-J. Bullinger, Editor. 1991, Elsevier Science Publishers B. V. p. 435-439.
[141] Yoo, J., Gervasio, M., & Langley, P. (2003). An adaptive stock tracker for personalized trading advice. Proceedings of the International Conference on Intelligent User Interfaces, pp. 197.
[142] Yoon, W. C., & Hammer, J. M. (1988). Deep-reasoning fault diagnosis: an aid and a model. IEEE Transactions on Systems, Man, and Cybernetics, 18(4), 659-676.
[143] Yoon, W. C., & Kim, Y. S. (1996). Aiding the analysis of human actions in large-scale systems: an intelligent interface approach. Computers and Engineering, 30(3), 569-577.
[144] Young, M.J. and Eggleston, R.G. (2002). Work-Centered Decision Support. In Proceedings of RTO Human Factors and Medicine Panel (HFM) Symposium held in Warsaw, Poland, 7-9 October 2002.

Acronyms

AI	Artificial Intelligence
AIA	Adaptive Intelligent Interface
BDI	Belief-Desire-Intention
CF	Canadian Forces
C4I	Command & Control, Communications, Computers & Intelligence
COGMON	Cognitive Monitor
COGPIT	Cognitive Cockpit
CommonKADS	Common Knowledge Acquisition and Design Structuring
DAI	Dynamically Adaptive Interface
EID	Ecological Interface Design
EMD	Explicit Models Design
HGA	Hierarchical Goal Analysis
IAA	Intelligent Adaptive Automation
IAH	Intelligent Adaptive Hybrid
IAI	Intelligent Adaptive Interface
IDEF	Integrated computer-aided manufacturing DEFinition
IISR	Integrated Intelligence, Surveillance, and Reconnaissance
MFTA	Mission, Function, and Task Analysis
OCD	Operator Centered Design
OAI	Operator Agent Interface
OMI	Operator Machine Interface
OWL	Web Ontology Language
PA	Pilot’s Associate
PACT	Pilot Authorizing and Control Tasks
PCT	Perception Control Theory
PRS	Procedural Reasoning System
SA	Situational Awareness
SASS	Situation Assessment Support System
SEAD	Suppression of Enemy Air Defences
TIM	Tasking Interface Manager
UAV	Uninhabited Aerial Vehicle
UCAV	Uninhabited Combat Air Vehicles
UML	Unified Modelling Langue
VACS	Variable Autonomy Control System
W5	What, Where, When, Who, and How

DOCUMENT CONTROL DATA
(Security classification of the title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (The name and address of the organization preparing the document, Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's document, or tasking agency, are entered in section 8.)

Publishing: DRDC Toronto

Performing: Nipissing Unversity, 100 College Dr., North Bay, ON P1B 8L7

Monitoring:

Contracting:

2. SECURITY CLASSIFICATION (Overall security classification of the document including special warning terms if applicable.)

UNCLASSIFIED

3. TITLE (The complete document title as indicated on the title page. Its classification is indicated by the appropriate abbreviation (S, C, R, or U) in parenthesis at the end of the title)

A Literature Review on Operator Interface Technologies for Network Enabled Operational Environments Using Complex System Analysis (U)
Une analyse documentaire sur les technologies d'interface opérateur pour les environnements d’opérations réseaucentriques à l'aide d'une analyse de systèmes complexes (U)

4. AUTHORS (First name, middle initial and last name. If military, show rank, e.g. Maj. John E. Doe.)

Haibini Zhu, Ming Hou

5. DATE OF PUBLICATION (Month and year of publication of document.)

May 2009

6a NO. OF PAGES (Total containing information, including Annexes, Appendices, etc.)

39

6b. NO. OF REFS (Total cited in document.)

142

7. DESCRIPTIVE NOTES (The category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of document, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Contract Report

8. SPONSORING ACTIVITY (The names of the department project office or laboratory sponsoring the research and development − include address.)

Sponsoring:

Tasking:

9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant under which the document was written. Please specify whether project or grant.)

13oc05

9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.)

W7711−083931/001/TOR

10a. ORIGINATOR'S DOCUMENT NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document)

DRDC Toronto CR 2009−051

10b. OTHER DOCUMENT NO(s). (Any other numbers under which may be assigned this document either by the originator or by the sponsor.)

11. DOCUMENT AVAILABILITY (Any limitations on the dissemination of the document, other than those imposed by security classification.)

Unlimited distribution

12. DOCUMENT ANNOUNCEMENT (Any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11), However, when further distribution (beyond the audience specified in (11) is possible, a wider announcement audience may be selected.))

Unlimited announcement

DOCUMENT CONTROL DATA
(Security classification of the title, body of abstract and indexing annotation must be entered when the overall document is classified)

13. ABSTRACT (A brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual.)

(U) A literature review was conducted to examine existing and potential advanced interface technologies for supervisory control of multiple heterogeneous assets (e.g., UAV swarming) in a NEO environment. These technologies include behavioural−based interface design approaches, physiological−based interface design approaches, and multi−agent interface design and implementation methodologies. The emphasis was on adaptive interfaces and intelligent agent system technologies. An analysis was conducted to compare differences between requirements of a NEO complex environment and current available technologies. A review on design principles and frameworks for synthetic complex systems was also performed. The results were summarized with pros and cons of different technologies for interface design purposes.

(U) Une analyse documentaire a été menée pour examiner les technologies d'interface de pointe existantes et potentielles permettant le contrôle de surveillance de nombreux biens hétérogènes (p. ex. groupes de VAT) dans un environnement d'opérations réseaucentriques. Ces technologies comprennent des approches de conception d'interface fondées sur le comportement et sur la psychologie, ainsi que des méthodes de conception et de mise en œuvre d'interfaces multi−agents. On a mis l'accent sur des interfaces adaptatives et sur les technologies de systèmes à agent intelligent, et on a analysé les différences entre les exigences d'un environnement d'opérations réseaucentriques (NEO) complexe et les technologies actuelles fournies. De plus, on a examiné les cadres et les principes de conception pour les systèmes complexes synthétiques. On a résumé les résultats avec les avantages et les désavantages de différentes technologies à des fins de conception d'interfaces.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (Technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus, e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

(U) supervisory control;UAV swarming;interface design; multi−agent interface design;complex environment;intelligent agent