AFRL-RH-WP-TR-2008-0002

AvantGuard: An Instrument to Explore Autonomy

Dov Jacobson
Stephanie Waugh
Jesse Jacobson
Thomas DiCesare
Edward Hobbs

GamesThatWork
Big Fun Development Corporation
620 Lakeshore Drive
Berkeley Lake GA 30096-3038

November 2007
Final Report for April 2005 to November 2007

Air Force Research Laboratory
Human Effectiveness Directorate
Warfighter Interface Division
System Control Interfaces Branch
Wright-Patterson AFB OH 45433

NOTICE AND SIGNATURE PAGE

TECHNICAL REVIEW AND APPROVAL

AFRL-RH-WP-TR-2008-0002

THIS TECHNICAL REPORT HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION.

FOR THE DIRECTOR

//signed//
GLORIA L. CALHOUN
Program Manager
System Control Interfaces Branch

//signed//
DANIEL G. GODDARD
Chief, Warfighter Interface Division
Air Force Research Laboratory

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
26 Nov 2007

2. REPORT TYPE
Final

3. DATES COVERED *(From - To)*
April 2005 - November 2007

4. TITLE AND SUBTITLE
AvantGuard: An Instrument to Explore Autonomy

5a. CONTRACT NUMBER
FA8650-05-C-6548

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
65502F

6. AUTHOR(S)
Jacobson, Dov
Waugh, Stephanie
Jacobson, Jesse
DiCesare, Thomas
Hobbs, Edward

5d. PROJECT NUMBER
3005

5e. TASK NUMBER
HC

5f. WORK UNIT NUMBER
66

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
GamesThatWork
Big Fun Development Corporation
620 Lakeshore Drive
Berkeley Lake GA 30096-3038

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Materiel Command
Air Force Research Laboratory
Human Effectiveness Directorate
Warfighter Interface Division
System Control Interfaces Branch
Wright-Patterson AFB OH 45433-7511

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RHCI

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-RH-WP-TR-2008-0002

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES
88 ABW PA Cleared 12/21/07, WPAFB-07-0767.
Report developed under Small Business Innovation Research (SBIR) Phase II Contract for Topic # AF04-071.

14. ABSTRACT
AvantGuard is a human factors research testbed developed with game technology and game concepts. It is used to study the effects of autonomy on a single operator supervising multiple unmanned aerial vehicles (UAVs). The synthetic task environment presents a fly-ahead convoy protection mission. Adversaries hide in the urban environment and prepare to attack. Using an innovative control interface, the operator directs the UAVs, studies their sensor streams, assesses danger and guides the convoy. AvantGuard provides a complex environment centered on four tasks: threat surveillance, threat detection, threat assessment, and threat avoidance. The level of autonomy (LOA) of each task is set independently from a broad range. The lowest autonomy levels demand full human attention. Middle levels offer limited operator intervention. The highest levels are fully automated, performed without the awareness of the operator. An interactive tool enables the researcher to sculpt autonomy profiles for comparative studies. A scenario development kit offers access to the synthetic environment. Each game-like session yields analytic performance metrics.

15. SUBJECT TERMS
Adaptive Autonomy, Levels of Autonomy, UAV, Autonomous Operations, Supervisory Control, Urban Reconnaissance, Game, Synthetic Task Environment, Human Machine Teams

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
UNCL	UNCL	UNCL	SAR	56	Gloria Calhoun
					19b. TELEPONE NUMBER (*Include area code*)

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI-Std Z39-18

This Page Intentionally Left Blank.

Table of Contents

Document Notes

In this document there are several abstract personalities.

Operator

This is the individual managing the proposed UAV system which AvantGuard simulates. The Operator (like the other parts of the AvantGuard system) does not exist now in real life.

Player

This is the real individual performing the role of Operator in a simulation.

Subject

This is the Player when considered as a source of performance data.

Researcher

This is a science worker who uses the AvantGuard testbed to collect Player performance data while controlling for certain variables.

Experimenter

This term is sometimes interchanged with Researcher, but it implies the personality who initiates the inquiry, designs the study and analyzes results. It may not be the same person who runs actual trials.

Designer

This is the individual who creates or modifies test scenarios. He supplies the environments in which the researcher can perform specialized tests.

Technician

This is a support member of the Research staff whose principle expertise is computers and file systems.

Developer

This is one of the workers who built the AvantGuard system. Every experiment performed on the AvantGuard testbed will be constrained by the limits of his imagination. It has been a design goal to reduce the influence of the Developer, and to deliver a system that responds readily to the curiosity of the Experimenter, the vision of the Designer and the experience of the Player.

Autonomy

This refers to the software agency that makes decisions in cooperation with the human supervisor. Frequently this function is treated as a localized agent: "The UAV replans its flight."

Sometimes it is a named subsystem: "The ATR detects a threat."Often it is treated as a generalized capability: "The Autonomy raises an alarm."

Pronoun

Where "he" is used to refer to any of these personalities, it is the gender-neutral pronoun.

The Balance of Autonomy

The AvantGuard testbed is a tool with which researchers can quickly evaluate autonomy alternatives for the human supervision of multiple UAVs (Unmanned Air Vehicles).

This is a critical area of research because the ratio of humans to aircraft is decreasing as the capabilities of autonomy increase. The role of the human in the next generation of UAVs will be better described as supervisor than as pilot.

UAVs and their human supervisors are in high demand as the mission of the Armed Services adapts to the challenges of asymmetric conflict. Intelligence, stealth, risk avoidance and realtime communication are as important as firepower. These requirements suit the capabilities of modern UAV systems.

Deployment of highly functional UAV teams demands careful design of the UAV autonomy. The attention and authority of the human must be divided among several UAVs and among the team's several tasks. If this collaboration is efficiently delineated, the system will fully exploit its human and computational talent. If not, they will impede one another.

The experience required to design such efficiency can be gained in the real world, while losing time, opportunity and fights. Or it can be developed cheaply and imperfectly at a laboratory testbed.

Figure 1: Threat-side view of AvantGuard

Human in the Loop

Unmanned Air Vehicles (UAV)

When a child loses his grip on a helium balloon and it escapes upward, that is an unmanned air vehicle. Everything else is manned. Aircraft do not yet fly without crews.

However the crew need not be on board.

UAV: Remotely Piloted

The plane can be remotely piloted. Some UAVs require nearly the same degree of control that a small plane requires of its pilot: able hands on the yoke, feet on the rudder. The UAV pilot flies by wire – as the pilot of a modern aircraft might. (Of course, he flies by wireless.)

The current enthusiasm for UAVs was boosted by the early dramatic success of Predators in reconnaissance and (later) combat roles in Afghanistan.

These planes carry no personnel on board, but they require a human crew. A functional Predator has a flight crew of pilot, sensor operator and (optionally) payload operator, mission coordinator or navigator.[i]

The full Predator system consists of four aircraft and employs fifty-five people. Similar crews man the Army's Shadow and Hunter UAVs.[ii]

UAV: Autonomous

Autonomy increasingly assumes the inner and outer control loops. It manages aerodynamic surfaces, avionics, navigation and basic flight planning. The Operator becomes less a pilot and more a director.

UAV Bonus

Almost every field enjoys the same benefits by introducing automation: robots work cheap, require little rest, and present no pension or morale issues. They accept tasks that are dull, dirty or dangerous. Labor cost, endurance and especially insensitivity to danger are prime benefits to military automation.

Aviation enjoys a special bonus. Without human support systems or safety standards, UAVs eliminate many physical constraints and greatly reduce the cost of production, maintenance and operation.

Supervisory Role

Historically: Many Humans, one Aircraft

Military aircrews can be as large as seventy people. Designers look to minimize manpower requirements by driving that number lower. The minimum has been one man per plane.

Basic UAV technology does not reduce this. Whether on board or telepresent, a pilot is required.

Future: One Human: Many Aircraft

It is not remote operation, but automation that changes this minimum. Even in manned craft, autopilots reduce manpower demands. But while the pilot can relax, he is still committed to that craft.

UAV technology combined with autonomy promises a single Operator flying a group of planes.

The Failure of Autonomy

Machine Failures

The shortcomings of all software are legend.

Autonomy

The shortcomings of autonomy are especially dramatic.

The DARPA Grand Challenge is a 150 mile race among Unmanned Ground Vehicles (UGV). A million dollar prize is posted. In 2004, fifteen engineering teams competed. Not one made it to Mile Marker 8.

Autopilots are more reliable. Machine vision is worse. Recent spectacular failures of machine vision software include face-recognizing crowd scanners prematurely deployed in airports after 9/11/2001.[iii]

AvantGuard can explicitly simulate certain machine failures, especially those of threat detection.

Human Failures

Machines are employed to extend human performance. Tasks selected for automation are generally those at which the machine performs better than the human. However the introduction of autonomy itself diminishes human performance. Scientists have catalogued several distinct forms of degradation:

Automation bias

Reasoning is impaired by reticence to disagree with the machine.[iv]

Complacency

Unconscious over-reliance is placed on automation.[v]

Workload Imbalance

Automation can force greater attention to lesser tasks.[vi]

Skill Degradation

Automation replaces human effort and skills are not retained.[vii]

Over-reliance

Excessive confidence is eagerly vested in automation.[viii]

Attention Narrowing

Decision-making is impaired by misdirected attention.[ix]

Mistrust

Unwarranted negative bias discounts machine decisions, based often on experience.

Impaired vigilance

A form of reliance is exacerbated by boredom, and increases over time.[x]

Adaptive Levels of Autonomy

Adaptive Systems

AvantGuard was first intended as a platform with which to experiment directly with adaptive autonomy systems. A technology of event sensing and adaptive response, feedback and hysteresis, was conceived to address the problem.

It became clear that this was premature.

Adaptive autonomy systems respond to a change in the environment by transitioning from an autonomy profile which had worked well to one which will work better under the new conditions.

Therefore an adaptive system cannot be designed until its target autonomy profiles are clearly established. These targets were not ready. The underlying studies have yet to be performed.

Researchers need to establish efficient autonomy profiles and match these to appropriate variables. With these in place, an adaptive autonomy strategy will fairly suggest itself.

Optimal Levels of Autonomy

The goal of the AvantGuard tool is to help researchers find the optimal autonomy profile for a system of human and UAVs performing a particular mission under certain sets of conditions.

When the gains of automation can most be realized without suffering a detriment in human effectiveness, the performance of the whole system is maximized.

Three Requirements

CONDITIONS

AvantGuard provides the Experimenter with a platform in which a great variety of scenarios can be realized to produce a variety of stressing mission conditions.

AUTONOMY

AvantGuard provides a tool with which the Experimenter can fashion a great many ($>10^4$) distinct autonomy profiles.

METRICS

AvantGuard measures the performance of Subjects working under given conditions with a particular autonomy profile. Metrics include an aggregate score and individual task-specific measurements.

Big Picture is composed of Small Details

Different autonomy profiles prevail under different conditions. With these three tools, AvantGuard can develop the data to map the relationship between variable conditions and ideal autonomies. Such a map would guide the design of an adaptive system.

Goals

There are several project goals in Phase I and Phase II.

1st: Level of Autonomy

The primary goal is clear: Deliver a tool to the Air Force Research Laboratory (AFRL).

It measures the effectiveness of a human and multiple autonomous agents in a simulated UAV mission.

2nd: Relevant Mission

Today real warfighters face real challenges in a hostile urban environment.

By simulating a difficult Military Operation in Urban Terrain (MOUT), AvantGuard may stimulate useful insights in creative tacticians.

3rd: Game Culture

Simulations and real control systems can gain by exploiting game culture and technology.

AvantGuard demonstrates the inherent value of game hardware, peripherals, software, interfaces and control conventions.

4th: Component Technology

In addition to the objectives listed above, an extra-scientific goal is mandated by the SBIR program. Research is funded only when it demonstrates commitment to a commercialization plan.

It is hoped that the AvantGuard testbed thrives in the military human factors research community. Marketing efforts are underway to promote its acceptance. Readers of this report are encouraged to contact its authors to discuss attractive AvantGuard applications. Some of these are discussed later.

Nevertheless, the market for new UAV research testbeds is quite limited. A prudent manager must defend the value of AvantGuard's component technologies. These components can address the needs of markets which are more broad and active than that of a specific research topic.

Such components may include the UAV control interface, or the terrain display system, and certainly the underlying game engine. Maintenance of these components as viable subsystems is threatened by a pressure to corrupt the architecture in order to expedite specific application results. The struggle to resist this pressure was manifest throughout development.

Software Product

AvantGuard is a complex software product. It includes a Player's interface to a game; a Researcher's interface to a testbed; a Designer's interface to a virtual world (see Figures 1 and 2) , and a Technician's interface to the digital structure.

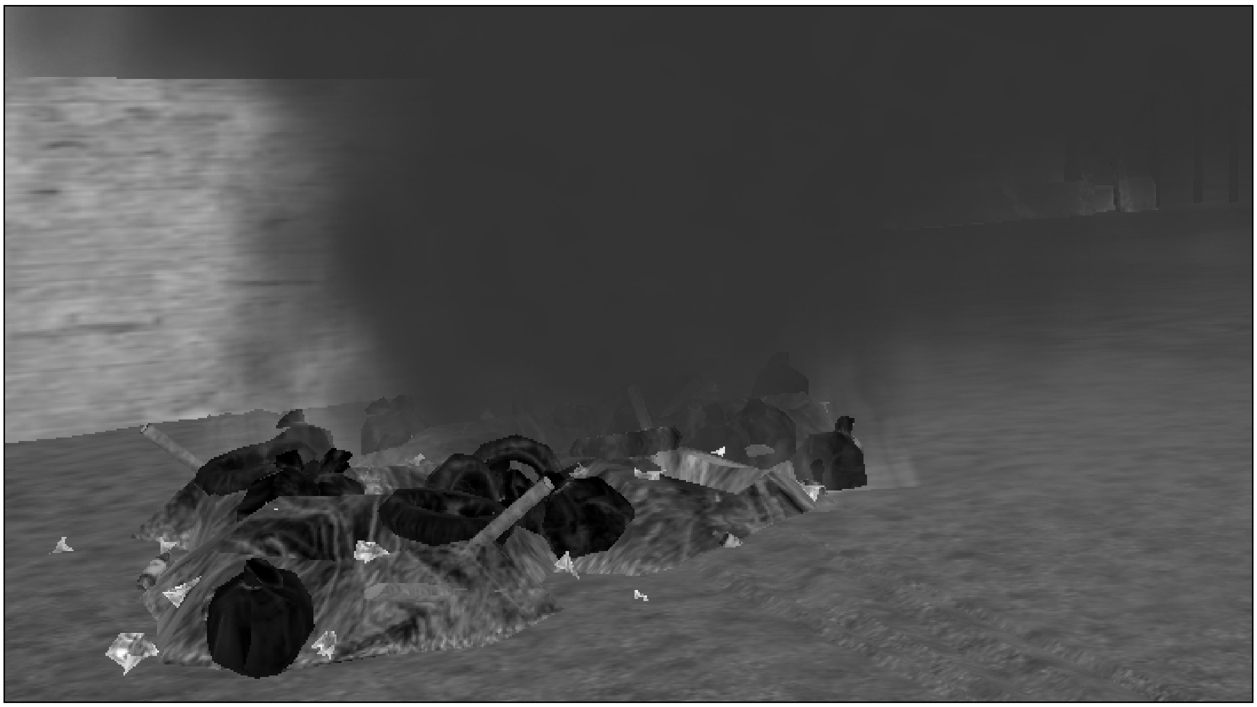


Figure 2: Burning Trash Pile

Game Elements

Mission

The Player is assigned to protect a lightly armed convoy which traverses unsecured urban territory. Using three autonomous UAVs, he must seek out threats, detect them when they are seen, determine their significance and route the convoy out of harm's ways.

The Player loses points if the convoy comes under attack by a threat.

Four Dimensional Display

Two Dimensions (2D)

Map

There is generally a map displayed on screen. Sometimes this is an embedded "mini-map" in the corner of the screen providing basic situation awareness. Sometimes it is a full screen map. The transition between the two is managed by the AvantGuard "Flywheel" navigation. (See ***INNOVATIONS.***) Most UAV simulations and control stations offer only this top down 2D view. In AvantGuard it is the starting point.

Video

Also inset into the screen are video windows that display the sensor stream from each of three UAVs. These are color coded to UAV symbols and UAV flightpaths on the Map and in the "Data Fusion" world.

Three Dimensions (3D)

Data Fusion

The main interface is a 3D synthetic view of the world. This includes accurate terrain, streets and featureless buildings. This might be a model of the town built by satellite or LIDAR telemetry. On this base model, the system records intelligence. Some is historic, some is automatic and most is the result of the human and UAVs cooperating in the mission.

Time

A Timeline runs across the bottom of the screen. Manipulating the Time Cursor allows the Player control over the time that is displayed in the Map, the 3D view and in all the UAV video windows. They all always show the same time. Events of significance (threat detections, for example) can appear on both the Timeline and the Map.

Predictive Display

When the Time Cursor is "scrubbed" into the future, the system projects the known entities along their current paths. Each Video window displays an estimate of the future view of the UAV. In the 3D Data Fusion view, the sensor's field of view is visualized as an illuminated volume.

Archived Display

When scrubbing backward, the system renders the world state which was recorded at that time.

Game Entities

Threats

In AvantGuard, the Designer can designate any object as a threat using the Model Framework to set the following threat properties. These properties are used by the threat detection system (Automatic Threat Recognition or Cuing) and by the scoring system to determine if the object looks like – or is – a threat.

Like all properties controlled by the Framework, these are dynamic. They can change as a function of time or another driving factor or in response to outside events. But they rarely have reason to change.

Threat Qualities

Any model can have three distinct threat properties.

Apparent Threat

This is the degree (0-1) to which this model looks like a threat. When it is within the camera's range of detection, this value is compared to the sensitivity of the Automatic Threat Recognition (ATR) system. If the threat appearance of the model breaches the sensitivity threshold, detection occurs.

Real Threat

This Boolean (true or false) flag informs especially insightful ATR profiles and helps score performance. It states – regardless of appearance – whether or not the model actually represents a threat.

Threat Class

This identifies the class of threat to which this model belongs. A half dozen classes are defined.

Threat Behavior

Independent of their listed properties, genuine threats incorporate hostile behavior.

Trigger

This Framework mechanism initiates an action when it is contacted by another object. Typically it is activated by contact with the convoy.

An IED threat, for example has a trigger in the road that initiates its explosion.

A more complex threat might react to an early trigger by running to an ambush spot and then respond to a later trigger with an actual assault. In AvantGuard the actual assault is rarely witnessed, since a UAV would need to be tracking the convoy at that moment.

Message

The visual behavior of attacking or exploding is interesting. However, a more important action occurs when the threat sends a message to the Game Manager declaring that the convoy was hit. The manager will penalize the convoy's "health" score according to the severity of the attack, which is generally a function of the threat class. When health is eroded to zero, the Game Manager ends the game.

Convoys

Convoys are complex entities with two components. Each has a list of members (Manifest) and a path it follows (Mission). The AvantGuard testbed is designed to support multiple convoys, but elements in the interface (the Timeline, particularly) assume only one convoy is being protected.

Manifest

The Manifest contains a list of vehicles (and/or pedestrians) in the convoy. It describes the convoy's speed and inter-vehicle spacing. AvantGuard ships with several different appropriate military vehicles and autos. The Designer can choose among these or add new ones.

Mission

The Designer (or the Player) can easily create Convoy Missions. These are street-bound straight-line paths from intersection to intersection.

AvantGuard supplies automatic convoy mission routing, based on the Dijkstra algorithm. It uses a threat map: a network of streets where each block has been assigned a threat value by either the human Player or autonomous software. Using this threat map, the autonomous system can plot a route that minimizes the cumulative threat exposure.

UAVs

The UAVs in the AvantGuard are relatively simple and generic. They evolved from characteristics of the Silverfox UAV (the AvantGuard simulation engine began on that project), but the Researcher can easily modify its specifications.

Flight Control Laws

These describe the aircraft's ceiling, its range of airspeed, acceleration, climb, roll and turn radius. They describe its sensitivity to turbulence.

These factors feed the flight simulation calculus, but they are not strictly observed everywhere. High fidelity simulation of a broad range of UAVs was never the goal of AvantGuard.

Sensors

The Researcher can modify the angle of view of the UAV's sensors. He can also mount the sensor on a fixed mount or on a slewable turret.

He can also determine the spectral sensitivity of the sensor. It can view:

- Daylight (EO)
- Night Vision (NVD)
- Infra-Red (FLIR)

Urban Features

Terrain

AvantGuard engine has a very detailed and robust representation of landforms. It can render real military DTED data or an imagined landscape. There is a rich system of weather, atmospherics and time-of-day effects.

All of these are readily changed by the Scenario Designer, following the Users' Guide.

Streets

There are geometric streets in AvantGuard scenarios. These are visible on maps and in the 3D Data Fusion view.

However, they are visually harsh. In the world seen by the UAV cameras, a more realistic painted road system is rendered while invisible geometric roads are employed by the logic and guidance systems.

Buildings

There are two types of buildings in AvantGuard.

Algorithmic

An algorithmic building generator creates typical Middle East architecture, with balconies, arcades, parapets and domes, as well as doors and windows. A special tool makes generation of buildings easy.

Modeled

A designer has freeform control when importing models to use as buildings. A good modeler will create much more efficient structures than the algorithmic buildings, and they may be more realistic.

People

Static

There are several posed models of local characters in static poses. These are the most efficient since they do not require the rigging (and expensive rendering) that animated models need.

Animated

Some models animate in a fixed location. People bargain in the marketplace. Couples chat on a balcony.

Guided

The most advanced people move around in the scene, giving the town a lively feel.

Vehicles

There are military, paramilitary and civilian vehicles, empty or occupied, parked or under way.

Urban Furnishings

The kit includes street signs (in Arabic), lamps, power poles and other mundane urban artifacts.

Autonomy

A central design challenge was to develop a system that allows non-programmers to carefully define subtle autonomy profiles and to provide a self documenting tool with which to do so.

The result was based on a simple design sketched in Phase I and developed in detail in Phase II.
The scope of design includes extremes of autonomy some of which await implementation in a later development phase.

Design

Strategy

The system autonomy is divided into four cognitive stages as described by John Boyd's OODA Loop[xi] (Orient, Observe, Decide, Act) and formalized by Raja Parasuraman[vii]. In AvantGuard, these four stages are:

THREAT SURVEILLANCE
THREAT DETECTION
THREAT ASSESSMENT
THREAT AVOIDANCE

For each of these stages, ten levels of automation are defined, strictly following the model by Thomas B. Sheridan.[vii] This model though admittedly arbitrary, is well accepted in the field. Compliance with it allows researchers easy comparison of AvantGuard results to the results of other studies.

During implementation, this rigid model was somewhat modified to accommodate reality. The real world (and even the highly simplified game world) is far more complex than the basic matrix of Parasuraman and Sheridan.

In general, this required additional new options for autonomy variation beyond the matrix's forty elements. These options are task-specific, and many are specific to a single autonomy level. They allow the Researcher to design numerous variations of each of the autonomy patterns.

Two features of the matrix, though well-defined were not implemented. These represented fine gradations at the extremes of the autonomy scale.

In each of the four stages, Levels 7, 8, 9 and 10 offer no different control to the Operator. They simply define the degree to which the system reveals its decision-making process to the human.

- 7: The machine decision-making is always shown.
- 8: It is shown when the human requests it.
- 9: It is shown when the machine determines that it is useful.
- 10: It is never revealed.

The logic of Level 9 is obscure in AvantGuard, which does not evaluate a condition in which the autonomy alerts the human but offers him no decision.

At the other end of the spectrum are extremely low autonomy levels. The autonomy scale has been biased to ignore the most manual methods. In some cases, AvantGuard's lowest autonomy (Level 1) might be Level 3 or 4 on another scale. Even still, potential users of AvantGuard will gain limited utility from the very lowest autonomy levels. These levels tend to specify tasks that require the full real-time attention of the operator. Such tasks are generally impractical in a multi-UAV system. Still, AvantGuard offers useful implementations for three of the four Level 1 autonomies.

However the Level 1 'Survey' task requires the Player to manually pilot all three planes. This was deemed unmanageable in multi UAV control and became a low development priority.

In the following pages, features not fully implemented in Phase II are marked with superscript circles (°).

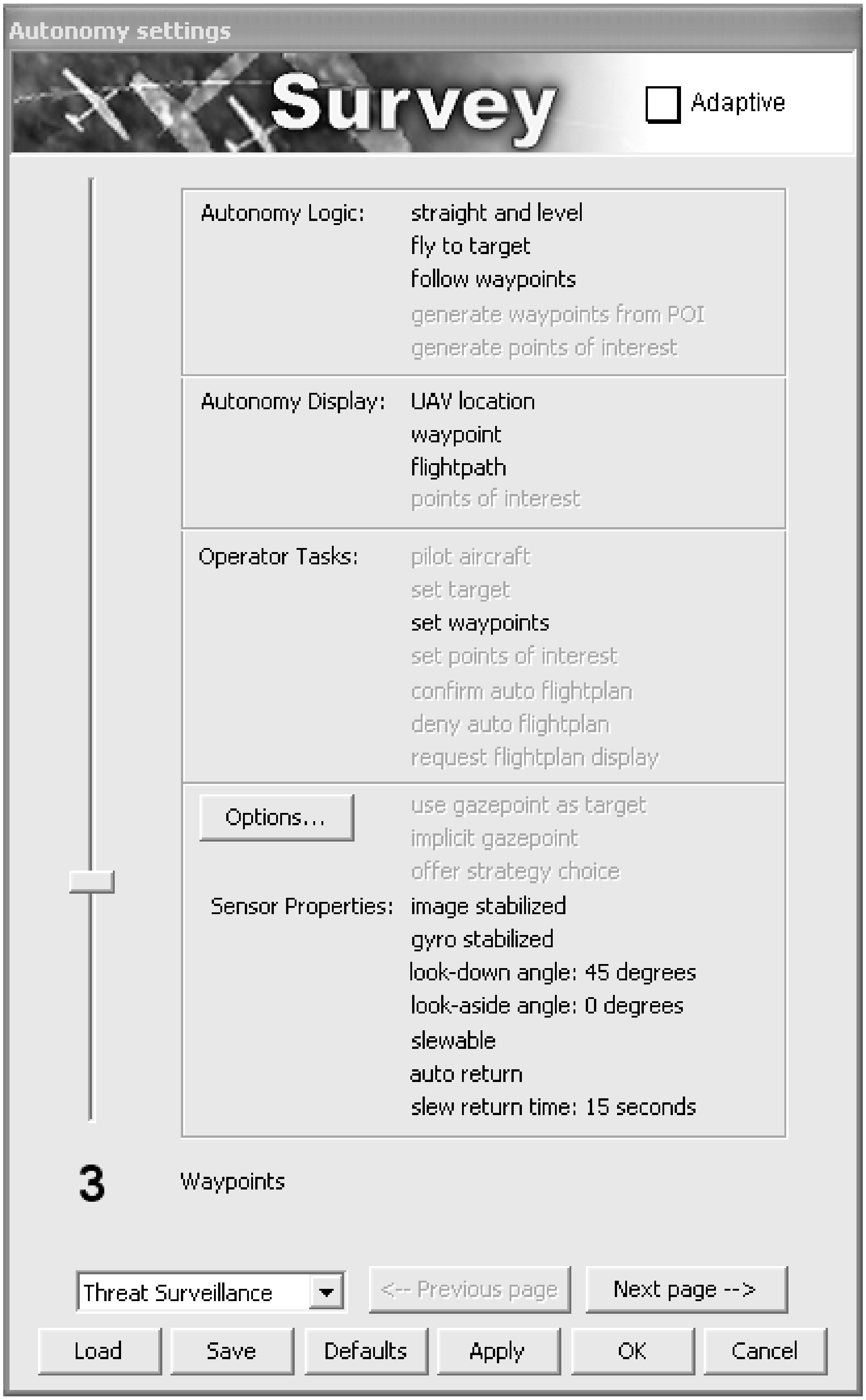


Figure 3. Autonomy Design Tool

Autonomy Design Tool

The autonomy levels are presented to the experimenter by a specially developed interface (Figure 3).

There are several elements that define the precise autonomy in use, and these are presented separately by the Autonomy Design Tool.

The four (Parasuraman) stages, and their (Sheridan) levels yield 10,000 potential basic settings and these are identified by a sequence of four numbers. Since Level 9 and Level 10 are never distinguished, the sequence can be conveniently coded as a 4 digit number (e.g., 3373) where each digit is between 1 and 9.

To establish these four levels, the interface presents the researcher with a slider. Raising and lowering the slider sets the numeric value. The interface also provides feedback to the researcher by displaying the precise features of each level.

As the slider is raised, more autonomous features are enabled and highlighted in the interface. As the slider is lowered, these are disabled and grayed out.

This display serves only to clarify to the Researcher the precise meaning of the levels. It does not permit the Researcher to set these features individually, since the basic autonomy features of all forty settings are established in the software design.

This display of level-determined autonomy features is divided into three sections.

Autonomy Logic

This section lists the steps of decision that the autonomy will perform. In general, these steps are cumulative. As the autonomy level increases, more of these steps are performed. This can be graphically seen in the interface: the list grows as the slider is raised.

Autonomy Display

In the classic model articulated by Sheridan, autonomy levels are often distinguished by the feedback they provide to the operator rather than simply by the actual tasks they perform.

Operator Tasks

If a decision-making task is not performed by the autonomy, it falls to the operator. This list of Operator Tasks complements the Autonomy Logic. As the Autonomy Logic increases, the Operator Tasks decrease.

This is seen graphically: A heavier Operator task load is displayed as autonomy decreases. However the named Operator Tasks generally describe the Operator's entire interaction with the system, rather than listing one element of his cognitive load. While Autonomy Logic displays a growing heap of cumulative tasks, the Operator Tasks tend to select from a menu of increasingly complex behaviors.

A concerned researcher will remember that the more difficult task (piloting the plane) subsumes the cognitive burden of the easier task (choose a direction).

Options

Ten levels at each of four stages suggest 10^4 different autonomy patterns. The additional Options increase this palette by another order of magnitude. Of course, only a small subset of these will be of practical interest to any investigator.

These Options are presented on a separate dialog from the slider. This dialog differs for each of the four stages, since each maintains a distinct set of task-specific options.

Even in the context of a specific stage, not all options are meaningful at every autonomy level. The interface makes this explicit. As the slider rises and falls, irrelevant options are grayed out.

Adaptivity

In each of the four stages, the “Adaptive” option opens a new interface. A novel interactive tool guides the Experimenter who wishes to program the adaptive behavior of this autonomy stage.

Adaptive transitions are modeled as a finite state machine. The Finite State Machine[xii] (FSM) concept was long ago borrowed from process control engineering by workers in the field of artificial intelligence. The FSM proposes that the agent is always in one of several well defined states. The detection of an event will cause the agent to migrate to another state. The new state defines new behavior, including its own set of event-driven state transitions.

The FSM is generally imagined by engineers as a two dimensional array, with a row for each of its states and a column for all potential events. Although this table defines the logic without ambiguity, it is usually sparsely populated and poorly visualizes the design.

AvantGuard’s innovative interface presents the Experimenter with a narrative perspective. Autonomy levels have an explicit escalating order. The Experimenter drags a trace along the screen. This indicates the active autonomy level. It ascends and descends at nodes which indicate events of interest to the Experimenter. (See Figure 4.)

This interface has been proposed and a tangible demonstration has been built and put in place. Actual functionality requires two elements of underlying technology: a system of event management and an autonomy controller driven by a finite state machine. These have not yet been developed and tested and are left as a goal for later development.

The tool has not been disabled, and remains a thought-provoking product of Phase II, as well as a frank invitation to continue this fruitful line of development. (See ***CONCLUSIONS AND RECOMMENDATIONS.***)

File System

Autonomy settings are preserved by the system by recording them to the scenario file. They are expressed in XML nodes whose format is documented in the Users’ Guide. These are aggregated into a single XML Autonomy Node.

The Autonomy Node can be embedded directly in the Scenario or it can be a named Autonomy setting included by reference. In the latter case, a separate file specifies the autonomy and is available to be reused in many different scenarios.

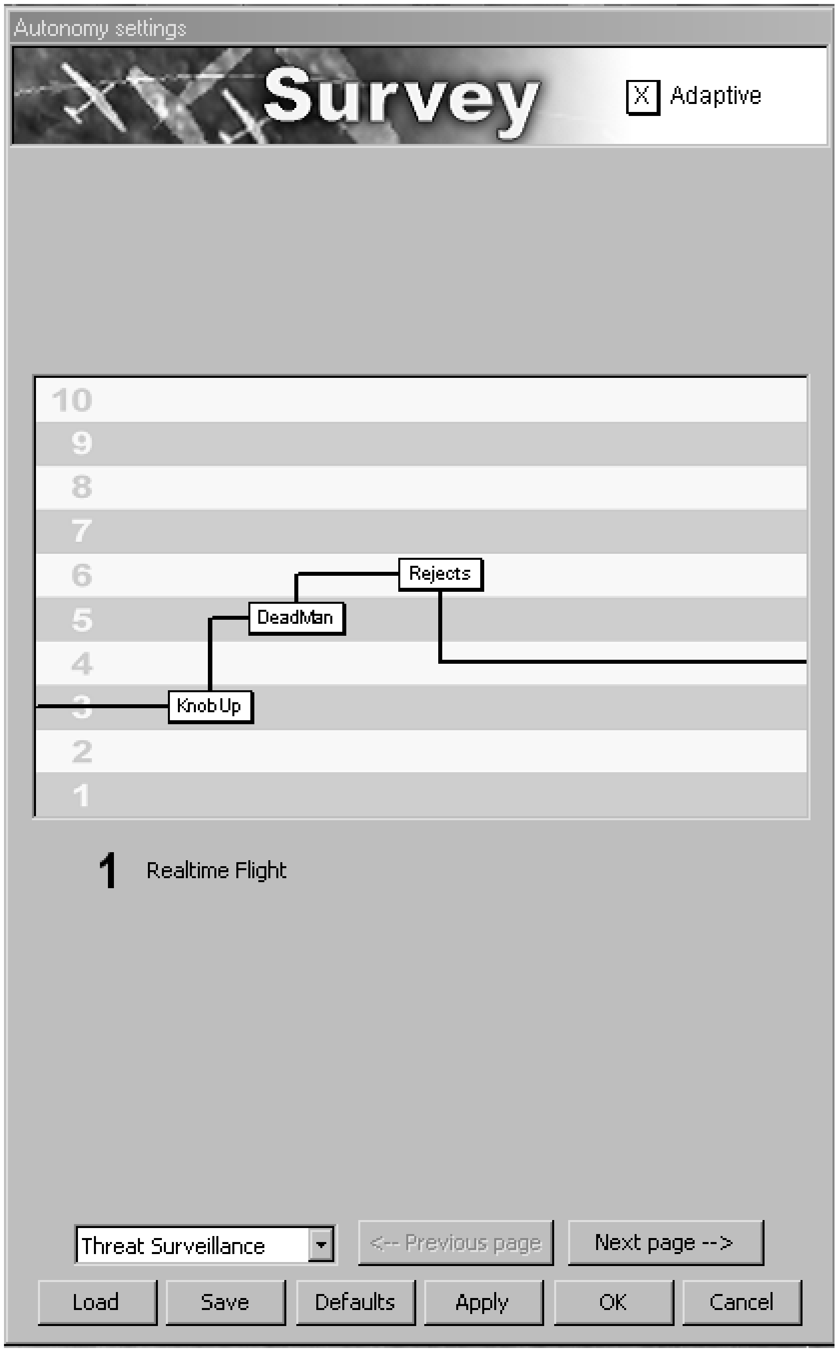


Figure 4: Autonomy Adaptivity Design Tool (Main Screen)

Threat Surveillance

Autonomy Logic

Straight and Level [Levels 1-10]

The autonomy always maintains the UAV flight control and keeps the plane in basic trim. The operator is never exposed to aerilons or throttle.

Fly to Target [2-10]

Autopilot will direct the UAV to the current target whether orbit point or waypoint.

Follow Waypoint [3-10]

The UAV follows a flightpath determined by consecutive waypoints. This flightpath may be set by the human or be autogenerated.

Generate Waypoints from Points of Interest [4-10]

This specifies that the flightpath is indeed autogenerated. It is based on a set of Points of Interest. This list can be supplied by the Operator or generated by the autonomy itself.

Generate Points of Interest [5-10]

This specifies that the autonomy will develop its own Points of Interest. Its logic is driven by the planned route of the convoy. Its goal is to get the maximize coverage of the upcoming convoy route.

Autonomy Display

UAV Location [1-10]

This determines that the UAV will be visible to the operator in the 3D Data Fusion display and on the inset Map. Each UAV is color coded to easily associate with its video window and its path and waypoints.

Waypoints [2-3]

In those autonomy levels where the plane is guided by waypoints, the autonomy may display the upcoming waypoints or none at all.

Paths [2-7]

A spline that describes the UAV flightpath can be shown to the Operator. This path will only show the future positions of the plane, not the path already traversed.

Points of Interest [4-7 (8 on request)]

Operator supplied Points of Interest can be shown. The autogenerated Points of Interest cannot be displayed.

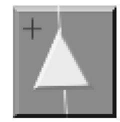
Operator Tasks

Pilot Craft °   [1]

The UAV is steered manually. The plane would follow a heading set by the operator.

Set Target   [2]

The Operator sets the immediate target for the flight. The UAV approaches and orbits this until a new target is set by the Operator.



Set Waypoints   [3]

The Operator manipulates a set of waypoints to determine the precise flightpath for each UAV. These can be dragged on the 2D map or in 3D and can be adjusted for altitude. Their orientation is determined automatically by the spline, which mimics the characteristics of flight.

Set Points of Interest   [4]

Using the Points of Interest tool, the Operator sets the targets which will be used by the autonomy to generate the flightpaths.

Confirm Auto Flightplan   [5]

When the Autonomy creates a new flightplan, it requests approval from the Operator. The timeout period for the decision is displayed to the Subject as a descending progress bar. He is also given a button with which he can review the original path and that proposed by the autonomy.

Deny Auto Flightplan   [6]

This reverses the dialog offered to the Operator by the Autonomy. A timeout in both cases produces the default result, but in this case the default is acceptance of the autogenerated plan. (At Level 5, the timed-out plan is discarded.) This level also reverses the sense of the preview button.

Request Flightplan Display   [8]

After Level 6, the Operator cannot change (or veto) the flightplan. However below Level 8, all flightplans are always displayed. At Level 7, the operator sees the plan, but can do nothing about it. Above Level 8, it is never shown. At Level 8, it is displayed when the Operator presses the button requesting display.

Options

Use Gazepoint as Target

Implicit Gazepoint °

The two Gazepoint features facilitate the Operator's task in Level 4. His task is to designate each UAV's immediate Point of Interest, which the UAV will then orbit and track. With Gazepoint, the Operator simply instructs the UAV to orbit and track the point he is looking at. This works best with a slewable camera. By pressing a trigger, the operator selects the point under the UAV camera's reticule.

Implicit Gazepoint suggests that the Operator need not press a trigger. Autonomy will sense when he is tracking a terrain feature. This was not made reliable before the conclusion of Phase II.

Experimental use of a Game Controller for Gazepoint control has been very satisfying.

Offer Strategy Choice °

Offer the Operator a choice between different approaches to the autogeneration of Points of Interest. In Phase II only one strategy is provided, so there is no consequence to this option.

Image Stabilized

Gyro Stabilized

These two stabilization features determine the degree to which the sensor imagery is isolated from the raw behavior of the plane. Some simple real-world UAVs have little such isolation. Consequently they deliver video that is nearly unwatchable. This seems unacceptable. If stabilization is not provided onboard, ground systems should provide it as a post processing feature. (In AvantGuard it does not matter whether it is considered to be onboard or part of the Ground Control system.)

Image stabilization removes the effects of turbulence. (The turbulence response model of each UAV is established as part of its aircraft profile.)

Gyro stabilization keeps the camera pointed level at all times. The horizon is always horizontal.

Look Down Angle

Look Aside Angle

These two features set the 'caged' orientation of the sensor camera relative to the plane fuselage. This is the angle at which a static camera is fixed, and to which a slewable/autoreturn camera returns.

Slew Return Time

Slewable

Autoreturn

These three settings determine whether the Operator can turn the camera in realtime (slewable) and if so, whether it returns to the 'caged' position after it is released, and if so how fast it returns.

Timeout

Apply to All

LOCKED TO 100 SECONDS IN PHASE II

This sets the timeout for Levels 5 and 6 when the autonomy requests Operator approval. If this countdown expires without Operator activity, the default behavior occurs. In Level 5, this means deny; or accept in Level 6. As a convenience, this interface will duplicate the timeout value universally to all other countdowns.

Threat Detection

Autonomy Logic

Cue Threat Detections [3-7]

At this level of Automation , the threat recognition system is engaged. The machine vision watches all three video streams and identifies suspicious artifacts. In the minimal implementation, Automatic Threat Cueing, it only notifies the player that a suspicious artifact has been observed. It marks these with a yellow circle on the video surface, and (optionally) on the Timeline and Map. It also enables the threat sightings statistics.

Maintain Playlist [4-6]

Threat Cues build up quickly in a dangerous environment and the Operator has a daunting task, trying to keep up with the flow of Cues. Clearly autonomy can take over the task of Threat Cue management.

The autonomy system maintains a list of unexamined Threat Cues. When the Operator examines a Cue, he has the opportunity to choose a response.

[View] Advance and maximize the video window to look at the cue.

[OK] Dismiss the Cue, after the Operator has decided whether or not to mark a threat.

[Prev] and [Next] leave the Cue in the system, while selecting an adjacent one for consideration.

Mark Threat [5-10]

Robust autonomy marks actual threats rather than simply presenting cues. The lowest level of Automatic Threat Recognition (ATR) (Level 5) is an upgrade from Automatic Threat Cuing (ATC). Both detect candidate threats and make tentative suggestions. In both cases, no threat is placed without the human's timely positive action. The difference is that the ATC only highlights suspicious imagery. All ATR levels designate a full threat marking with class, location and timestamp. More autonomous levels of ATR (Levels 6-10) insert threats into the Threat Map without Operator approval.

Autonomy Display

Realtime Video [1-7 (8 on request)]

In all but the highest autonomy levels the Operator sees the video stream transmitted from the sensors in the UAVs. At the very lowest level of autonomy, this raw live feed is the only resource available to the Operator with which he can detect threats.

In the highest levels of autonomy (Levels 9, 10 and sometimes 8) the video feed is omitted and the interface is radically different. At these levels, the human is expected to trust the ATR system to detect and mark any threats that fall within the field of view of the UAVs. The Operator is not expected (and in Levels 9 and 10, not permitted) to second-guess the autonomy by reviewing the video source.

Timeshift Video [2-7 (8 on request)]

Only in the lowest autonomy is the Operator required to keep eyes-on the video feed at all times and to make instant recognitions.

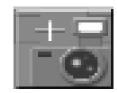
The timeshift (TiVo™) function allows the operator to review the video archive with instant replay, slow motion and scrubtime, while the UAVs fly and record new video.

Operator Task

Mark Threats [1-4]

The essential task of threat detection is to mark (designate) threats. While the video is the source of information, the marks are not being placed in the video but in the 3D world that the video represents.

More precisely, in fact, threat designations are placed in 4 dimensions. They are located in the world at a particular moment in time. This stage produces only points of detection. The next stage (Assessment) extrapolates these four dimensional points to predict future danger.



Control video displays [2-7]

With the added assistance of TiVo™ comes the work of managing it.

Without TiVo, the Operator is under pressure to detect threats, knowing he is responsible to see what is on all three screens at all times.

TiVo, arguably, only increases the pressure. At any time, the Operator is able to view everything that has ever passed beneath the UAV sensors. His responsibility is actually increased. He can no longer forget about missed threat sightings. (See ***SUGGESTED STUDIES***)

Examine Cued Threats [3-6]

The introduction of Automatic Threat Cuing in Level 3 elevates the Operator's task. He no longer monitors the video source uniformly. Instead he considers the segments deemed suspicious by the autonomy. He serves as an image analyst, authorized to make mission critical recognition judgments.

Confirm Autonomous Detection [5]

In this classic implementation of Autonomy (Level 5), a functional ATR system detects, places and classifies the threats and tentatively marks them in the 3D map.

However these are not registered in the system unless positive confirmation by the Operator is received before the timeout limit expires.

Deny Autonomous Detection [6]

While Level 5 requires Operator confirmation, Level 6 requires Operator denial. Passive Operators permit the autonomy to determine the threats.

Request Video Display [8]

At Level 8, the Autonomy is in nearly full control. There is no video display and no Timeline. The only action that is available to the Operator is to turn the video on so that he can see what the ATR system is seeing when it detects threats. Even this weak assertion is taken away in Levels 9 and 10. (In Levels below 8, the video is always available and this task is not relevant.)

Options

New sightings only in range °

Slip time to expand range °

Archive Video Available °

These three options represent ambitions that exceed the limits of Phase II.

Of these, Archive Video is the most interesting and most ambitious. It refers to the concept of making available to the Operator the stored video recorded at this location on missions days and weeks earlier.

The experience of veterans propelled this concept. A British Army Officer, reflecting on fifteen years of patrolling Northern Ireland, suggested that the clues of impending trouble are frequently not positive features (the lurking gunmen) but rather absences or changes in established patterns (e.g., no children playing in the street at 1630). Archived video offers the Operator a long baseline for such observations.

The technical challenges were great. Innovative display technology was developed by the engineering staff. Archive video files can be retrieved from the file system and played.

However serious functionality requires a video database system to be developed – and then populated. Much effort would be required to create a database system based on location and time. And a far greater effort would be required to fill the system with video content to represent months of history of the town. Such efforts exceed Phase II limits.

Sightings on Map

Sightings on Timeline

These two options control the appearance of unconfirmed sightings. These are autonomous detections not yet confirmed or denied by the Operator. They include both managed Threat Cues (Level 4) and tentative Threat Marks (Levels 5 and 6).

Sightings made by the Autonomy generally appear in the 3D Data Fusion world. They will also appear in the Map, and on the Timeline. The Researcher has the ability to suppress these displays.

The symbol on the Map is a projection of the Data Fusion icon. In Data Fusion, it has three dimensions, but on the Map there is no height. Except in the case of rooftop snipers, this is harmless data reduction.

On the other hand the Timeline icon adds information to the Data Fusion icon. The two are meant to be used together. The Timeline supplies an easily visible *t* dimension to the *x*, *y* and *z* of the Data Fusion.

Fidelity

Sensitivity

Noise

These three options control the quality of the threat detection engine used by the ATR and ATC.

The quality of autonomy is a critical factor when evaluating levels of autonomy. If we consider an automatic system whose decisions are made perfectly, then the best level of autonomy will always be 100%. (If this is not true, our quality metrics are faulty.)

In fact, the automatic system need not even be perfect. If it is simply better than the human alternative, then complete autonomy is appropriate.

Sensitivity controls the threshold at which an apparent threat triggers a threat detection event. It assumes a system in which the image analysis function produces a figure of merit between 0 and 1, where 0 indicates a completely innocent appearance and 1 is an unmistakably threatening figure.

Fidelity allows the Researcher to activate a special feature which distinguishes appearance from reality. Unlike traditional automatic threat detection systems, which rely on analyzing imagery from simple sensors, our detector can peer directly into the soul of any individual and ask: Are you really a threat?

It can query objects that do not traditionally have souls, such a truck or a pile of refuse that seems to be booby-trapped. Even a dead dog, whose soul (if it ever existed) departed days ago, will honestly report whether its carcass hides an IED.

When Fidelity is 0 (default) the autonomy will not use its soul-searching ability, but will behave like real-world ATR and judge things by their appearance alone. With Fidelity set to 1, the appearance is ignored and the ATR works like a stern angel, identifying sinners even before they have committed any act.

A fractional value blends the two methods of judgment. Appearance predominates at values below 0.5, and intent above.

Threat Detection Simulation

At this point it would be useful to explain the workings of the AvantGuard threat detection system:

Real threat detection systems perform image analysis to isolate features within the sensor stream. They then subject these features to pattern classification systems to match against known threat profiles.

Such machine vision makes extreme demands on computational resources, and returns dubious results in the real world. Real machine vision has no place in a simulation which already fully employs every processor cycle. It would use more power if more were available. It cannot share the CPU with another process whose demands are equivalent.

Nor is there need to analyze the synthesized imagery when the world model is directly accessible.

Each camera that supports ATR looks into the world and identifies those objects which lie within its field of view. It then examines the properties of each object to determine if it appears to be a threat, if it is a real threat and to what class of threat it belongs.

This requires prepared data. In AvantGuard, the Scenario Designer must evaluate the threat appearance, the real threat potential and the threat class of any entity which will excite the threat detector.

Threat Appearance is a fractional number, where 0 is an innocent appearance, and 1 is completely alarming. The threat detector compares this to its own current sensitivity. If the object’s threat appearance exceeds the detector’s sensitivity threshold, detection occurs.

Notice that there is no random element here. Even when imitating faulty ATR, the system will produce the same results for all sessions. This makes it a more reliable testbed instrument. It imposes a burden on the research team, who must contend with the fact that subjects quickly learn a predictable scenario. To defeat this, the Researcher requires many alternative but equivalent scenarios.

Noise is the opposite method of introducing failure into the autonomy. With one click, it can introduce considerable replay value to any scenario. However, this method introduces testing uncertainty and has been locked out of recent builds of AvantGuard.

Sorted queue °

When active, this option presents the Threat Cues to the Operator in order of immediacy. They are sorted by time-on-target (i.e., how many seconds until the convoy encounters this threat).

Threat Assessment

Autonomy Logic

Interpolate Sightings [3-10]

When multiple sightings have been made of the same threat at different places and at different times, a good autonomic system can connect the dots. It will not only display the threat sightings that have been reported but will estimate motion between these sightings. This is considered the first step up from an entirely manual system.

Generate Threat Map [4-10]

More robust autonomies go beyond interpolation of marker locations. They integrate the multiple threats that converge at each point to complete a total threat map.

Predict Future Threats [5-10]

This is similar to Interpolate Sightings, except that it connects the last known sighting and the presumed goal of the threat. Determining that goal is non-trivial. Currently the Player must suggest the goal.

Project to Goal ° [3-8]

In this case the Player does not suggest a goal and the responsibility to do so falls to the Autonomy.

Autonomy Display

Threat Markers [1-7]

Threat Markers, produced by the Detection stage are the input of the Assessment process. When they are displayed the Operator can review the autonomic logic and satisfy himself that good assessments are being made. Threat markers appear in the Data Fusion View (3D) and on the Map (2D) and optionally on the Timeline.

When the input data is hidden, the Operator has no choice but to trust the machine.

Compress to Single Threat Marker [2-7]

If this feature is active, the autonomy will consolidate separate sightings of a single threat in order to track and predict its movement.

Scrubtime/Realtime Threat Matrix [3-6]

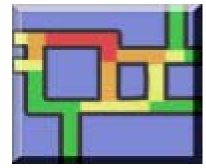
The Threat Matrix accumulates the threat potential of every block. It displays a color-coded threat advisory map which changes over time. Generally this map is synchronized with other parts of the system display. Time is controlled by the Timeline and its scrub bar.

Time on Target Threat Matrix [7-10]

This is another type of threat matrix. This map does not show the scene at one moment in time. Instead it shows a range of time. Each block displays its threat potential at the moment when the convoy will arrive. (See ***INNOVATIONS.***)

Operator Task

Assign Threat Levels [1-3]



When the autonomy does not do so, the human must estimate the cumulative effect of nearby threats on each block and resolve this to a block-by-block threat level.

Street threat levels have one of five values: green, blue, and yellow, orange, red.

They all start on blue and the human must assign any other threat level using the street painter tool.

Predict Threat Directions ° [4]

In order to help the threat predictive function, the human provides a guess as to the threat direction. So far this has been unworkable.

Initiate Regeneration [4]

At Level 4, the operator decides when to refresh the threat levels with an auto-generated threat matrix.

Confirm Autonomic Threat Projection [5]

When the autonomy creates a projected assessment, the human must determine that it is acceptable.

Deny Autonomic Threat Projection [6]

The human has a chance to reject any autonomic threat assessment.

Display Threat Map [8]

The Operator is not shown the threat markers upon which assessment is based, but can request them.

Options

Bind Threats to Streets

When this option is set, threats travel only on the streets.

Show on Timeline

Show the threat sightings on the Timeline.

Use Real-Time Threat Matrix

Always show the threat map at current time (i.e., extrapolate detected threats to the present).

Use Scrub-Time Threat Matrix

Always show the threat map as it was – or will be – according to the current scrub time as set by the Operator using the Timeline.

Use Time-on-Target Threat Matrix

Regardless of both the current time and the scrub time, show each point in the threat map at the time of convoy arrival. It assumes the convoy travels by the most direct route at full speed.

Threat Matrix Properties

The Researcher can easily assign a weight to the risk presented by each class of threat.

Unknown

These streets have not been examined. This value represents the average danger in this town.

Lookout / Cameraman

A suspicious character with a cell phone and/or video camera but no immediate weapon.

Veterans of real missions have noted that the presence of a video cameraman is often the most salient clue of a planned detonated IED attack. The cameraman is an essential member of the attack team. He must have a good line of sight to the ambush point and cannot stand off as far as the bomb detonator.

Small Arms

Each sighting represents an individual with gun or rocket propelled grenade. A force of multiple people is represented by multiple markers.

IED

This marks a direct sighting of a suspected IED. This is generally disguised in a heap of trash, in a discarded sack, even in the carcass of road kill.

Vehicle IED

This represents a highly mobile and destructive threat. It is a truck with explosives. The same class doubles for a truck or car filled with armed men.

Historical Threats

Here is the weight which the Autonomy should attach to previous incidents known to occur in this area.

Threat Avoidance

Autonomy Logic

Follow Direction [1-2]

At the minimal autonomy available in the system, the convoy retains one immediate instruction, such as "Turn Left". It will execute this instruction at the next intersection.

Raise Alarm [2-4]

At this low autonomy, the convoy will monitor the Threat Map and alert the Operator before danger occurs. It is the Operator's decision whether to respond to the danger. The actual convoy route re-plan may be performed by either entity.

Reroute [4-10]

The Convoy autonomy will maintain a path to its destination. At low levels, the Operator inserts detours around the threats. At higher levels, threat-sensitive Autonomy creates safe convoy routes.

Initiate Reroute [5-10]

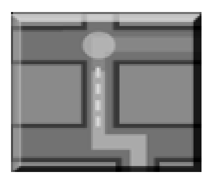
At high levels, the Autonomy decides whether a new threat map requires a new route.

Autonomy Display

Display Convoy Route [3-7]

Generally the Operator will be shown the Threat Map used to plan the Convoy route. At high levels of autonomy, he is not shown this. (In Level 8, it is available but not displayed by default.)

Operator Task



Direct convoy [1-2]

When the autonomy is low, realtime attention is required of the Operator, who is responsible for every turn made by the convoy, and must issue timely directions. These are: "Left" "Right" and "Straight".

Reroute Convoy [3]

Although this task is well understood and can be reliably automated, it is not always automated. The Operator builds (turn-by-turn) a convoy path that avoids the worst dangers in the threat map.

Initiate Reroute [4]

This declares that the Operator must monitor the convoy's current route on the threat map, in order to initiate a new plan when the present one becomes too dangerous.

Confirm Autonomic Reroute [5]

The Autonomy cannot proceed with a reroute until the Operator approves. He is given a button that lets him toggle the display between the old and new routes.

Deny Autonomic Reroute [6]

This is the same, but a passive Operator implicitly accepts the decision of the Autonomy.

Request Route Display [8]

At Level 8, the Operator will see the current convoy route only when he explicitly requests it.

Options

Voice Control °

With this option enabled, the Operator speaks his turn-by-turn directions of the convoy rather than indicates these with button-presses. This feature was introduced early in Phase I, but it has since been neither removed nor maintained. It serves more as a demonstration feature than as a solid testbed element.

Audible Alarm °

Visual Alarm °

These two options determine the method by which Autonomy alerts the Operator of impending danger.

Threat Sensitivity

This slider allows the Researcher to adjust the strategy of the Threat Avoidance autonomy. A low sensitivity produces the minimum-length route regardless of threats, while a high sensitivity results in risk-intolerant, evasive paths. (At a sensitivity of 1.0, the routing logic is oblivious to threats.)

Look-Ahead Time

This determines how far into the future the autonomy should look in order to raise alarm.

Metrics

The development of performance metrics was aided by the cooperation of researchers at the University of Tennessee (Knoxville)[xiii].

Data Product

Session Identification

Each session has a serial number and a Subject name. It is date and time stamped. It identifies the files which specify the autonomy profile and the scenario details being studied.

Timeline Usage

Three statistics measure the Subject's use of the Timeline. They record the proportion of session time which the Subject spent reviewing past events, examining future plans or operating directly in realtime.

Direct Interaction with Autonomy

THREAT SURVEILLANCE

Five metrics track the Subject's interaction with the aircraft guidance autonomy on those levels (Levels 5 and 6) where direct dialog is enabled. They measure how many of the candidate flight paths suggested by the autonomy were approved by the Subject, rejected by the Subject, allowed to time out by a passive Subject or were never seen by a Subject who was not ready. (The last case is that in which the dialogue is suppressed by an earlier pending dialog.) It also tracks the Subject's use of a feature that enables him to alternate the display between the two options between which he is asked to select.

THREAT DETECTION

Five metrics similar to those of Threat Surveillance measure the Subject's interaction with Threat Detection autonomy. However it includes additional direct interaction at Level 4 (Automatic Threat Cues). Note that Cue dialogs are never suppressed; they are managed in a queue.

THREAT ASSESSMENT

Metrics similar to Threat Surveillance are applied to the autonomous production of Threat Maps.

THREAT AVOIDANCE

These five metrics track the Subject's acceptance of autonomous convoy re-routes.

Performance of Threat Detection

The system of threats and their detection yields five performance metrics that track the number of threats seen versus the number in the world and the number overflown by the UAVs.

Performance under Low Autonomy

Three metrics count the number of times that the Subject performs manual tasks: planning UAV flights, establishing new Points of Interest, or new Convoy routes.

Scores

These metrics report an overall score based on avoiding contact with the threats, as well as the number of objectives within the scenario and how many of these are accomplished.

In Session Reports

The system sends messages to the Subject as directed by the Scenario and by standard events. While all such messages are preserved in log files, the most significant are also recorded here in the results table as concatenated text.

Data Format

Purpose

The initial data formats produced by AvantGuard emphasized the detailed chronologic experience of each session. They were designed for analysis in pattern classification systems that might seek to characterize individual sessions or even meaningful sequences within a session. This draws from the contractor's previous experience supplying pattern classifiers in a very different problem space. For the insights into human factors that AvantGuard is seeking to discover, pattern classification analysis is an unlikely direction and a bad influence on data formatting.

Long freeform chronologies fought longitudinal analysis. Metrics that produced narratives could not readily feed the statistical mechanisms designed for analyzing bulk experience.

Form

The end result is a simple design, which makes up in usability for what it lacks in sophistication.

All events are accumulated, and reported as counts per session. Every session is a fresh row in a table whose 43 columns are invariant. The data is stored in simple CSV (Comma Separated Value) format.

Format

CSV was chosen for easy import by spreadsheet software which typically can read it directly. It is also easily integrated into the workflow that builds a relational database or a commercial statistical analysis package.

Trial management

Each session produces a row in the currently active table. This corresponds to a results file in the AvantGuard file system. On any system, only one table is active at any time. The Researcher can initiate a new table or reactivate an existing table by a simple interaction with the AvantGuard console. This information is preserved across sessions and system reboots.

Every session has a unique serial number. It is unique across tables, but it is unique only on its native machine. The Experimenter has full control over this and can increase its uniqueness by assigning a separate range to each machine – or conversely by resetting the serializer for each table.

Data Preservation

When the tabular rather than narrative metrics were chosen as the useful data product, this imposed an early data reduction which was considered valuable. It is worth noting that the chronology data is still preserved in densely detailed system logs. Ambitious data miners can extract the story of each session and apply sophisticated analysis techniques in search of new insights.

Project History

Phase I

Prototype

The Phase I effort resulted in a functional prototype. Although it was incomplete, it demonstrated a playable game that explored the interaction of a single human and multiple autonomous UAVs.

ADAPTIVE

It was early in Phase I that the goal of the project stepped back from direct creation of an adaptive system to creation of a tool that can design target autonomy profiles which such an adaptive system might ultimately employ.

PHOTO CHIPS

The Phase I prototype borrowed tasks from existing exemplars.[xiv, xv] The main activity revolved around managing large amounts of still photographic imagery while contending with limited pixel bandwidth. The Player could only view a small fraction of the sensor imagery that was captured.

Phase II

Direction

Phase II began with a commitment to develop the prototype seen in Phase I into a full research tool with adaptive autonomy and a simple playable game. This was well understood and manageable.

Ambition

As the project staff became more familiar with the state of the art, they considered a redesign. Instead of a system based on still photography, they considered one based on video. This would make AvantGuard more relevant to actual practice.

Feature Gallop

ANIMATION

The challenge was irresistible. At that time the Works engine was quite primitive. It did not include animated figures. This was seen as the major impediment to switching to a video based test bed. Animated figures require much engineering, but since it's a feature every engine needs, why not now?

TIMELINE

Animation was the tip of the iceberg. It required the Works engine to catch up with the state of the art. But video introduced far more demanding issues. Some demanded genuine technology innovation.

The Phase I photo based system always simulated the present. When the Player reviewed the past, he viewed actual snapshots that the system had recorded in the past, and which it displayed in the present. It is not possible to record video and review it this way. Video requires too much bandwidth, especially in a system that displays three video streams at all times.

MULTIPLE SIMULATION

A unique system was developed that allowed the Player to scan backward through the video while the simulation moved forward. This required the engine to support multiple simulations simultaneously. At the same time it calculates the current state, it also displays a previous state. (See ***INNOVATIONS.***)

REWORK: BAD AND GOOD

The extra engineering and content creation placed a much larger burden on the project than expected. Nevertheless, a more sophisticated AvantGuard emerged from this process. The rework aligned AvantGuard more closely to real world UAVs, and it gave rise to a much cleaner autonomy model.

Testing

A critical element in the success of a testbed is testing. The Phase II effort offered limited opportunity for laboratory testing. Instead, AvantGuard was refined in a regular series of on-site demonstrations and open critiques where the project benefited from the experience and creative input of many Laboratory scientists.

Meanwhile the AvantGuard testbed was subjected to trial under laboratory conditions during a parallel project. This study was a Synthetic Task Environment Workbench[xvi] at the University of Tennessee (UT). This study required test results from a multiplayer version of AvantGuard.

This project provided an opportunity to field test AvantGuard in a laboratory which is part of the Mesa Research Site University Consortium. AvantGuard developers refined many of its features in this environment. They added network support and redesigned the data product.

Deliverables

Software

The software, its tools and its data product are described here, and in more detail in the documentation described below.

Documentation

The core documentation is the reference *Guides for Users*. A separate guide is delivered for each class of user: Player, Researcher, Designer and Technician.

The documentation also includes *Step-by-Step* tutorials for several important operations.

Training Scenarios

Outside the scope of the contractual deliverables, the contractor has produced a series of narrated, training scenarios which serve as interactive tutorials to prepare test subjects. This serves to give each Subject a uniform introduction to the software, and it lightens the training burden for the research staff.

Technical Report

This document.

Future Development

Software is never finished. Near its end, every software development project enters a phase during which ambitious features are pruned away because they are incomplete. In some cases they will never be revived. In other cases these ambitions must bide their time, waiting for a new round of funding.

Hopefully AvantGuard's are in the latter group. Several exciting features have been planned – even explored- that could be delivered inexpensively while the project is fresh, and the team is still focused.

Scenarios

AvantGuard invites scenario development by the research staff. Nevertheless, professional outsourced scenarios might accelerate AvantGuard's adoption. See ***SUGGESTED STUDIES.***

TRAINING

Five training scenarios introduce new Players easily to AvantGuard. These can be extended.

EXPERIMENTS

Scenarios can be crafted to support experiments designed, conducted and analyzed by scientists.

Adaptive Levels of Autonomy

The design of the adaptive system is quite clear. The interactive tool for creating adaptive sequences is already presented as a prototype and its paradigm can be explored by researchers. A development effort to realize the tool's functionality would involve very little risk.

Autonomy

EXTREMES

The Autonomy Matrix still has boxes unfilled. Most are extremes which are of marginal relevance. But exploration of AvantGuard's autonomy space may stir interest in more extreme features.

DETAILS

Within each level of autonomy are several options. Some are subtle, others are significant. At the end of Phase II most of these options were operational, but a few were not implemented, or implemented only as demonstration features.

Speech-based control can be developed more fully. Turbulence can have a more complex model.

Game Controller

It is a project of modest scope to fully integrate the Game Controller into the AvantGuard interface.

Moving one's point of view and picking out threats are familiar videogame tasks. They will be easy to implement, pose very little risk and will surely provide satisfying player experience.

Other challenges (interface navigation, joystick text entry) have familiar workaround conventions.

The most difficult is the controller-based substitute for fine-control dragging. Moving UAV waypoints, rerouting the convoy and tweaking the threat map are all likely to require development and tuning.

Innovations

SBIR contracts are meant to tap the innovative capabilities of the independent R&D community.

The Phase I and Phase II AvantGuard efforts yielded new ideas. AvantGuard demonstrates several features which may contribute to the evolution of the interface between UAVs and their human supervisors. These novelties include:

Pilotless Perspective

Bidirectional Timeline

Zenith Map

Flywheel

Homing Cam

Superimposition Cam

Time On Target Threat Map

Game Controller [partial]

Pilotless Perspective

Remote Cockpit

Previous UAV control systems evolved from the perspective of the pilot. Popular ground control interfaces reproduce cockpit instrumentation on computer screens.

AvantGuard was developed with little input from pilots. The interface will be used by one person performing a critical task using several UAVs. If these UAVs do not already have highly automated flight functions, the game is already lost. Even a skilled pilot is not expected to fly three aircraft simultaneously.

Intelligence Center

If the Operator is not a pilot, who is he?

He is an intelligence officer. He is the sensor operator on the Predator, or the “MI Guy” on the Army’s Hunter. He is not overly interested in the oil pressure readings. He is more interested in data about the terrain he is studying than in details of the aircraft that carry the cameras.

Data Fusion Screen

The main AvantGuard screen depicts the terrain in an abstract 3D representation. There is free movement through the world, which includes the ability to zoom outward and upward to see the world as a map.

Information collected during the mission is embedded into this scene, and (although this is only lightly explored in AvantGuard) it can be integrated with intelligence developed from outside sources.

Collection, analysis and extrapolation of this data is the focus of the AvantGuard screen, and the aircraft are treated like videogame planes – simple reliable entities which can be directed to perform within their capability envelope but whose inner workings are someone else’s problems.

Bidirectional Timeline

A central element of the AvantGuard interface is the Timeline that runs along the bottom of the screen. It reinforces awareness of this as a problem in four dimensions, of which time is often the most critical.

Present

Usually, the Operator is monitoring the real world and the Timeline is displaying realtime. The cursor progresses from left to right as possibility is reduced to history.

If the Operator presses the 'Pause' button, this moment will freeze. Video screens, Map and Data Fusion display all stop moving. However the simulation continues - undisplayed. The planes move unseen and transmit unmonitored video streams. This is as it would be in real life.

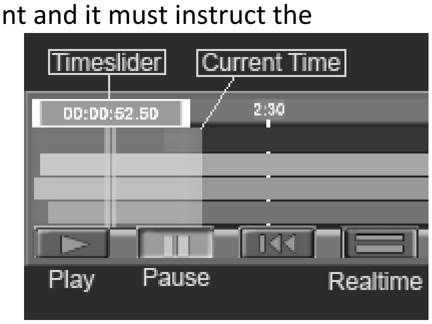
Past

Of course, when the present moment is frozen, it immediately joins the past. AvantGuard signifies this on the Timeline. The Time Cursor lags behind as the Current Time marker progresses forward.

The Data Fusion world becomes sepia-tinted to indicate that the past is displayed. By dragging the Time Cursor (scrubbing) further to the left, the Operator looks back further in time. Pressing 'Play', replays the archived video at real speed. Pressing 'Realtime' returns to the present.

This simple functionality was very difficult to achieve. The real life TiVo™ process is relatively straightforward. It records incoming video while replaying the video from an earlier point in the stream.

The heart of the game engine is not a video player, it is a simulation. The simulation must always calculate the present world and its interactions and consequences. If it is displaying the past, it must reconstruct that past simulation at the same time as it maintains the present and it must instruct the cameras to render the correct reality.



This was a difficult (and apparently unique) technical achievement in the simulation sciences. It is not unknown for a videogame or simulation to record a session and replay it at the end. The achievement is to do so while the real simulation progresses forward. The engine must maintain multiple overlapping realities.

Future

If we can cast the simulation into the past , why not also cast forward into the future?

Here the system is not replaying past facts, but projecting current assumptions forward. The entities whose plans are known (the UAVs and the Convoy) will march forward in the simulation. This aids the efforts of planners to coordinate and to deconflict.

It gets more interesting as the system attempts to project forward the entities whose plans are unknown – in particular the threats that have been identified in the world.

Integration

This integrated sense of time - where the fleeting present separates an unknowable future from an unchangeable past – seems to be a uniquely tangible feature of this software.

Situational Awareness Tools

Maintenance of Situational Awareness (SA) is always an issue in simulation, games and actual military operations. AvantGuard introduces several features that are meant to reinforce Situation Awareness.

Mini-map

The most traditional of these is the mini-map. This small map represents a larger territory than does the main view. The extent of the main view is generally indicated on its surface. This has been a feature of videogames since it was introduced by *Defender* in 1980.[xvii]

Zenith Map

When the Operator views the 3D world from directly overhead, it transitions into a 2D top-down map. The switchover begins at a given distance, and transitions smoothly and quickly. During this transition, the mini-map fades away (it is redundant). Zooming in on the Map returns the view to the 3D world.

Iso Cam

Learning from strategy games like Blizzard's *Warcraft III*, and from popular applications like *Google Earth*, AvantGuard introduces an Iso Cam. The features of this camera are designed to balance flexibility, speed of motion and solid SA. Experimenters can work to improve this balance, by modifying its features, which are easily accessed. These features include:

Flywheel

The mouse wheel is called the Fly Wheel in AvantGuard. It integrates control of zoom, elevation and pitch. Rolling the Flywheel forward and back flies closer and further. As the camera flies back, it also rises up and points down. This keeps the same features in the center of the screen.

FLY OUT: Zoom back and rise to a bird's eye view. Zoom back further into the Zenith Map.
This exponential motion typically takes 2-3 seconds to go from close-up to map view.

FLY IN: Zoom toward an object and arrive at eye level.

North Facing

It improves SA to ensure that the Player's view (in 3D and in the Map) always faces North.

SELF RIGHTING

The Player can still rotate the view, so that it no longer faces North. When he releases the control, it pivots back to North-facing.

Superimposition

When one of the UAV video feeds is selected (e.g., to more easily mark threats) the camera automatically moves to superimpose the video window over the synthetic view. This reinforces SA. The dynamics and trajectory of the camera have been refined, when early versions were found to erode SA.

POV Illuminator

The area of the view for each UAV is shown in the Data Fusion world by a headlight, which duplicates the camera view and shines into the world. It illuminates what the sensor will see.

Time-on-Target Threat Map

Threat Maps

A goal of the threat assessment stage is to translate the individual sightings of threats (whether by human or automated threat detection) into a large picture of the threat environment.

In AvantGuard, this picture is a map of all the streets in the town, with each block assigned a threat advisory color:

- Blue: if it has not been surveyed
- Green: if it has been surveyed - and no threats were found
- Yellow, Orange and Red: increasing number and lethality of threats

The threat map typically represents the state of the environment at a particular point in time.

This was implemented in two ways.

One shows the Realtime threat map. It displays the current location of all known threats.

The other shows the Scrubtime threat map. The Timeline's Time Cursor sets the threat map to the same point of time that the videos and main view map are displaying.

Using the latter method (scrub time threat map), the Operator can scrub forward and observe where the convoy will be in the future and what threats will be encountered.

But there is a better way:

Time on Target Threat Map

One of us (Jesse Jacobson) developed an autonomous map which looks into the future, but not at a single point in time. Rather it displays a large range of the future at once.

At each intersection on the Map, the system uses Dijkstra's algorithm to calculate the arrival time of the convoy. In Phase II, this is based on the shortest route from the convoy's current position.

The threat map generator evaluates the threat at each intersection at the time of arrival of the convoy.

Beyond UAVs

This kind of map – especially if it were maintained autonomously – could be of significant value in many military situations by helping commanders visualize the future.

For example, the 1993 disaster in Mogadishu resulted from a moving ambush which flanked the Americans on both sides. The attackers traveled faster than the convoy down parallel streets, and met the convoy again at every intersection.

A simple threat map would show the threat localized at one intersection. A Time-On-Target Threat map would more accurately display a mile-long gauntlet of crossfire.

Game Controller

An initial investigation explored operation of the AvantGuard testbed with a Game Controller.

Some of the functionality was quickly converted and a system became operational enough to reveal that it would be a useful interface, and a worthy subject on its own for human effectiveness studies.

Interested observers have 'play-tested' the controller in its limited application. It provides a capable and appealing interface. Among younger project engineers, it is the preferred UAV sensor slew mechanism.

Deferred Implementation

This project was introduced late in Phase II. Initial exploration showed that full implementation requires a serious effort. Play-tests show that it warrants this effort. Full integration of the Game Controller is an excellent goal for later development.

Suggested Studies

DISCLAIMER

The proposal of actual psychological experiments is outside the scope of this project and outside the authors' expertise. But it might be useful to consider some potential applications of this software.

The following are three exemplar studies that might be easily performed with the AvantGuard testbed. They illustrate the value of the testbed to provide data that may help answer important questions.

These sample topics include basic human-machine psychology research and practical usability research. The usability research aims to support designers as they contemplate future UAV systems.

The Impotent Observer

Background

Sheridan's analysis[vii] of Levels of Automation puts a large premium on simple feedback. Of his celebrated scale, a full forty percent of the range is devoted to autonomy schemes in which the Operator is completely unable to affect the autonomic behavior. But these levels (7, 8, 9 and 10), are differentiated by the degree to which the Autonomy reveals to the Operator its decision-making process.

Hypothesis

The Operator's performance will be impaired when the Autonomy hides its reasoning.

Test Bed Preparation

Scenario

The Operator's task load concentrates on reassessing the threat map in response to a stream of threat detections arriving from the ATR system. When the Map shows that the convoy is headed into danger, the Operator must reroute it safely and quickly.

A scenario that requires several convoy reroutes would be useful. The UAV flights and the threat placement can be arranged so that threats are revealed just a short distance ahead on the convoy's current route. The route is too dangerous. The Operator must have a clear idea of the alternative paths and choose one immediately.

Autonomy Profiles

The autonomy pattern ***8723*** will cause the UAVs to fly independently. Their paths and their locator icons are hidden from the Operator. They mark threats completely automatically. The Operator must monitor these popup threats on his map and reroute the Convoy accordingly.

The pattern ***8823*** will change the display so that the Operator does not see the video streams at all. He only sees the threat markers supplied by the ATR. This might simulate UAVs without video transmitters or with non-visual sensing systems.

Metrics

The Operator is shown all the threats and merely has to guide the convoy around them. This is his sole job. A reading of the convoy health (an inverse measure of threat hits) will be a good metric of job performance. It will also be possible to measure if and when he chose to turn on the UAV video.

Retest

If there are significant results from this test, it may be worthwhile to perform the test again, reversing the roles of Autonomy and Operator, and using ***7278*** and ***7277*** autonomy profiles.

The Burden of TiVo™

Background

The development team is proud of a few innovative features that might influence future UAV system designers. High on this list is our timeshift capability. (Bending trademark laws, we call it here "TiVo".)

It allows the Operator to perform instant replay while the simulation marches forward. It allows him to fully review all decisions he made (or deferred) earlier. The assumption is that more information and more flexibility are inherently helpful to the decision-maker.

Perhaps our assumption is wrong. Perhaps the availability of the replay actually undermines the confidence needed to make firm and fast decisions? Maybe the added responsibility distracts him? Its use certainly competes for time and attention.

Similar questions are often asked about the influence of instant replay on sports referees.[xviii]

Hypotheses

At a sufficiently high workload, the ability to review past decisions degrades overall performance.

If reviewing past decisions degrades current performance, the Subject will cease reviewing.

Even if is not used, the ability to review past decisions will degrade performance.

Test Bed Preparation

Scenario

Researchers can set up the testbed to intensify on the threat detection task. The subject can be required to monitor diverse video feeds with many threats hidden among ambiguous distracters.

If the UAV missions are preset, the timing of the threat appearances can be carefully scripted. Threats can appear in two video feeds simultaneously. The UAV paths can be designed so that the alternate routes that the convoy must take to avoid threats are well explored by the UAVs. Perfect performance of threat detection will avoid any trouble in the automated assessment and avoidance steps.

Autonomy Profiles

The autonomy pattern ***1179*** will insure that UAVs are impossibly difficult to reroute, and that Assessment and Avoidance are highly automated. This pattern can be tested against a pattern of ***1279*** which only adds the TiVo functionality.

Metrics

If the design of the scenario is carefully executed, so that the UAVs overfly all the significant threats, the only cause for threat encounters will be failures of detection, so convoy health is a good metric of task performance which will serve to score hypothesis 1 and 3. Hypothesis 2 can be measured directly by tracking the use of the Timeline as reported in the results data package.

You Don’t Know What You Don’t Know

Background

Donald Rumsfeld’s prescient phrase illustrates a danger that newcomers experience as they enter an alien environment. This is a particular disadvantage associated with asymmetric conflict in which an advanced and very public force contends with a hidden improvisational force on the latter’s terrain.

Superior technology is a key advantage on which the more developed force is prone to rely. While the technology itself offers an advantage, reliance can be detrimental. Unaware of the scope of the new terrain, the newcomers are likely to see it through lens of this technology.

They will have a tendency to accept the view presented by the technology. Assuming that the new world is already in front of their lens, they will not really drive their sensory equipment. They will allow it to drive them.

Hypothesis

Practical constraints imposed by technology stimulate self-imposed constraints on strategic imagination.

Test Bed Preparation

Scenario

This scenario is designed to induce complacency, then to exploit complacency and to harshly punish it.

All three UAVs follow the same flightpath at staggered times. It is the convoy route. The UAVs patrol the route intensely to make sure it is clear for the convoy. The Operator is taught how to change the paths of the UAVs. Unless he does so, he will explore only this one route. Along the route, there are interesting details, but no threats, apparent or real.

Suddenly a disturbance flares up along the route, one block in front of the convoy. The unready Operator reacts quickly and diverts the convoy down an attractive alternative – and directly into a trap.

Autonomy Profiles

An autonomy pattern of ***4474*** requires the Operator to initiate a change of UAV flightpaths (by selecting new Points of Interest) and it assists detection with Automatic Threat Cuing. It requires the Operator to direct the Convoy and will raise an alarm if there is an apparent threat in the path. The only difference between the test and control groups is that one has fixed, forward looking cameras and the other has control of (and training in) slewable cameras. (This is an Autonomy option, within Threat Surveillance.)

Metrics

Explorative Subjects are likely to be aware of the trap. Tunnel-vision Subjects will not. Score will be a good metric. A count of how many times the UAV flights are replanned would be especially valuable. Ideally AvantGuard would also count the number of times the camera is slewed and seek a correllation.

References

[i] **KURLE, D. (MAJ, USAF)**
Predators Provide Eyes In The Sky Over Afghanistan
Air Force News Service; Jun 09, 2006

[ii] **COLUCCI, F.**
Schoolhouses for UAV Pilots Up and Running
National Defense Magazine; May 2004

[iii] **SCHEERES, J.**
Airport Face Scanner Failed
Wired; May 2002

[iv] **SKITKA, L. J.; MOSIER, K. & BURDICK, M. D.**
Accountability and automation bias
International Journal of Human-Computer Studies; 2000

[v] **KNAPP, R. K. & VARDAMAN, J. J.**
Response to an automated function failure cue: An operational measure of complacency
Human Factor Society 35th annual meeting, San Francisco, CA; 1991

[vi] **YEH, M. & WICKENS, C. D.**
Display signaling in augmented reality: The effects of cue reliability and image realism on attention allocation and trust calibration
Human Factors; 2001

[vii] **PARASURAMAN, R.; SHERIDAN, T. & WICKENS, C.D.**
A Model for Types and Levels of Human Interaction with Automation
IEEE Trans. Systems Man and Cybernetics; May 2000

[viii] **WICKENS, C.D.; CONEJO, R. & GEMPLER, K.**
Unreliable Automated Attention Cueing for Air-Ground Targeting and Traffic Maneuvering
Proc Human Factors and Ergonomics Society the 34th Meeting. Santa Monica, CA; 1999

[ix] **AHLSTOM, V.; LONGO, K. & TRUITT, T.**
Human Factors Design Update –Automation Guidelines
DOT/FAA/CT96-01; Feb 2002

[x] **MOLLOY, R. & PARASURAMAN, R.**
Monitoring an automated system for a single failure: Vigilance & task complexity effects
Human Factors; 1996

[xi] **BOYD, J. (COL, USAF)**
Discourse on Winning and Losing
Unpublished briefing series, 1970-1990

[xii] **BLACK, P.E.**
Dictionary of Algorithms and Data Structures
U.S. National Institute of Standards and Technology. 24 February 2006.

[xiii] **JACOBSON, D; WOEHR, D. & BOWLER, M.**
Synthetic Task Environment / Workbench: Final Technical Report (Phase I)
Air Force Research Lab, Human Effectiveness Directorate July 2007

[xiv] **TSO, K.; THARP, G.; TAI, A.; DRAPER, M.; CALHOUN, G. & RUFF, H.**
A Human Factors Testbed for Command And Control Of Unmanned Air Vehicles
22nd Digital Avionics Systems Conference; 2003

[xv] **BARBATO, G.**
UCAV Controls and Displays for Suppression of Enemy Air Defenses
CSERIAC Gateway; 2000

[xvi] **STTR CONTRACT FA955007C0019**

[xvii] **ORLE, K.**
A Brief History of Videogames
IGJA Wiki

[xviii] **BODLEY, H.**
Umpires do decent job, don't need instant-replay help
USA Today; 24 Oct 2005