*The 18th Digital Avionics Systems Conference*
St. Louis, MO, October 24-29, 1999.

# A Multi-Agent Operator Interface for Unmanned Aerial Vehicles

Kam S. Tso, Gregory K. Tharp, Wayne Zhang, Ann T. Tai
IA Tech, Inc.
10501 Kinnard Avenue
Los Angeles, CA 90024

## 1 Introduction

Future unmanned aerial vehicles (UAVs) are designed to fly at long range and high endurance, and to perform complex and demanding tasks such as surveillance and reconnaissance, electronic warfare, command and control, target acquisition, and weapon guidance. The current preprogrammed ground control strategy will not be able to meet the task complexity, long range and high endurance requirements. In this paper we describe a multi-agent operator interface developed to resolve the challenging issues in controlling future UAVs, including:

- Sensor payload management in support of a wide variety of mission types (e.g. coordinating use of camera images, terrain maps, data from meteorological, radioactivity, chemical agent sensors and detectors).
- Complexities of flight operation, navigation, target spotting and acquisition, weapon deployment, and operation of a variety of mission payloads.
- Collaborative planning and coordinated control of multiple UAVs by an engaging unit in achieving a common mission goal.
- Real-time responsiveness given perturbations to connectivity between the air vehicle and ground control station.

The building blocks of the Multi-Modal Immersive and Intelligent Interface for Remote Operation (MIIIRO) is shown in Figure 1. The foundation of the MIIIRO operator interface system is a distributed architecture which is built upon World Wide Web and distributed object technologies such as the Java Remote Method Invocation (RMI) and Common Object Request Broker Architecture (CORBA) [1]. The Web-based architecture allows client software running on the MIIIRO stations to be deployed over the network and executed on the ubiquitous Web browser. The distributed architecture provides flexible and secure communication between the MIIIRO stations and their server for information collection and distribution. It also allows users separated by large physical geographical distances (e.g., Commanders at the Pentagon, Field Commanders at the mission center, and crew members in the battlefield) to participate interactively in situation assessment and collaborative planning.

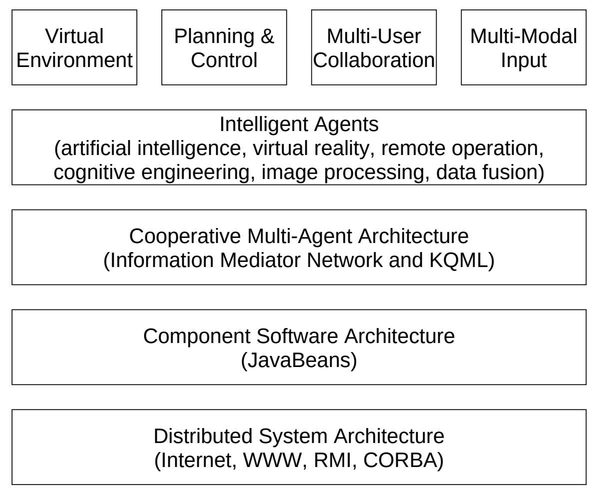


Figure 1: Building Blocks of the MIIIRO System

The MIIIRO software architecture is highly modularized so that it can be easily reconfigured to meet new requirements and expanded for future applications. The MIIIRO agents are implemented using the JavaBeans component technology. Each agent is an independent light-weight specialist with a well-defined interface. The MIIIRO system is assembled from these agents, and some may even be purchased from the JavaBeans component marketplace. The component interface of JavaBeans makes it easy for new components to be added and existing components to be enhanced and replaced in the MIIIRO system.

The MIIIRO cooperative multi-agent architecture facilitates communication and coordinates activities among the intelligent agents. The architecture is based on the *information mediator network* [2] which classifies agents into different functional groups and manages each of them with a special agent called Task/Information Manager (TIM), as shown in Figure 2. The TIM supports cooperative problem solving, activity management, and intelligent information delivery. The architecture has many advanced features and benefits. First, a new agent can be easily "plugged" into integrated operations by connecting it with a TIM. This makes the architecture open and scalable if new functionalities and more agents need to be added in the future. Second, group TIMs can simplify information management and significantly reduce communication overhead. The cooperation and interaction among two agents in the same group can be resolved by the group TIM. Furthermore, with connected TIMs, a complex action plan can be composed from the subplans of relevant TIMs, each of which is dedicated to carry out its own subplan as efficient as possible. This in turn can significantly increase the overall system performance.

The distributed system architecture, component software architecture, and cooperative multi-agent architecture provide a flexible, reconfigurable, and expandable foundation for the intelligent agents to assist the operators in planning and controlling the UAVs. The interface of the JavaBeans software components ensures reconfigurability and expandability at the *system* level. The cooperative multi-agent architecture ensures reconfigurability and expandability at the *task* level by coordinating the intelligent agents, implemented as JavaBeans components, to work as a whole so that a sequence of actions can be carried out towards a common goal.

A large set of intelligent agents, embodying the knowledge of diverse disciplines, have been developed to provide the planning and control, multi-user collaboration, virtual environment, and multi-modal input capabilities for achieving MIIIRO's goal of being an intelligent assistant cooperating with operators in planning and controlling UAV missions.

In the past, virtual environments are typically proprietary and expensive to acquire and maintain. The COTS products and technologies used in this project allow us to develop a low-cost multi-modal immersive operator interface for UAV control. Because the software is developed in Java and VRML which are both platform independent, the resulting system can operate on a low-cost, light-weight PC running the standard Windows 95/NT operating system. This feature is critical for the battlefield environment where troop mobility and component interchangeability are critical.

# 2 System Architecture

## 2.1 Distributed System Architecture

The distributed system architecture, which is shown in Figure 3, is built upon the Internet, World Wide Web,

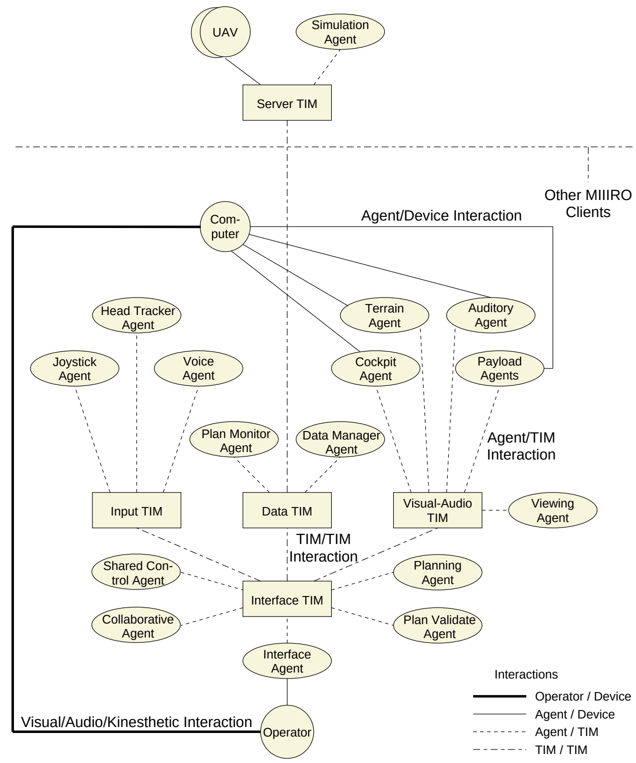


Figure 2: MIIIRO Intelligent Agents

CORBA, Java and VRML. All these technologies conform to open standard and are platform-independent, thus allowing MIIIRO to be run on most hardware platforms. An additional benefit is that there is no need for software deployment since the Java applet and VRML models are downloaded to the Web browser.

Within this architecture, the operator at a MIIIRO station uses a web browser, such as the Netscape Navigator, to download the MIIIRO client software from an HTTP server. The MIIIRO client software consists of the 3D models of the terrain and UAVs specified in the Virtual Reality Modeling Language (VRML), and the intelligent agents implemented as Java beans packaged as a Java applet. The Web browser invokes the Java Plug-in to execute the Java applet and render the 3D models. The operator interacts with the virtual environment and the graphical user interface of the Java intelligent agents to generate plans and commands for controlling the UAVs. The commands are forwarded to the MIIIRO server which relays them to the UAVs via satellite communication. Data from the UAV flight instruments, camera images, and mission payloads are sent back to the MIIIRO server which stores them in the global database and also forwards them to the MIIIRO stations.

To accomplish a military mission, MIIIRO will need to access heterogeneous information from various sources, such as the Geographical Information System for aerial

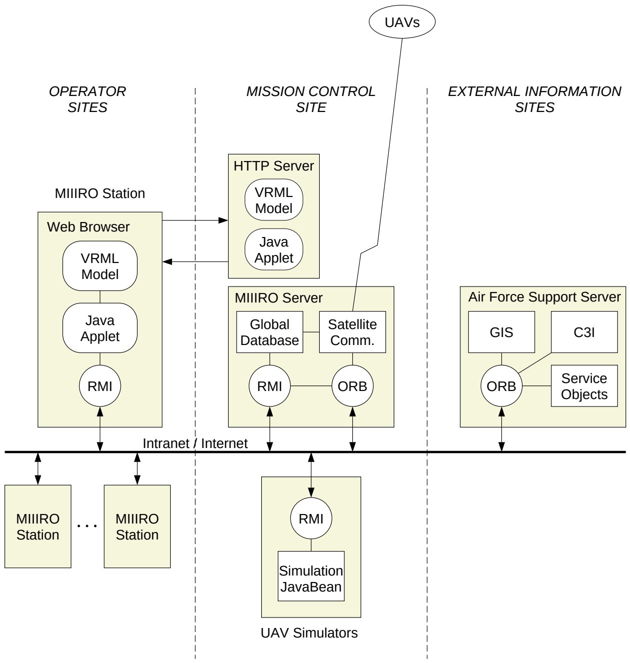


Figure 3: Architecture of the MIIIRO System

maps, C3I support for battlefield intelligence and friend-or-foe discrimination. The Object Request Broker enables network wide access to heterogeneous information systems.

The HTTP server and MIIIRO server can be hosted on the same computer. The servers can support many MIIIRO stations at the same time, only limited by the load generated by global database access, command and status network traffic, and satellite communication bandwidth. The load on the MIIIRO server is minimized since the Java applet and VRML model are locally cached by the Web browser, and a local database is maintained at each MIIIRO station.

The MIIIRO stations, MIIIRO server, and the HTTP server can be co-located in a UAV mission control center and connected by high-speed local-area network such as 100 megabit Ethernet or an ATM network. The architecture also allows MIIIRO stations and the MIIIRO/HTTP servers to be located at different geographical locations, such as running the MIIIRO stations in the battlefield while the servers are located at a control center miles away.

## 2.2 Cooperative Multi-Agent Architecture

The MIIIRO system will be operated in a distributed, unreliable, and chaotic environment. The UAVs, MIIIRO stations, and MIIIRO server are separated by large geographical distances connected by unreliable satellite links, radio links, and networks. The battlefield environment where it operates is both dynamic and chaotic. MIIIRO stations are shut down and brought up from time to time due to the movement of the engaging unit, and connection of Commanders from different locations. In such constantly changing distributed environments, it is imperative to have a cooperative multi-agent architecture which can keep track of all the resources in the system and can coordinate the agent activities.

In this project we adopted the information mediator network [2] for organizing and coordinating the agents. The information mediator network consists of individual agents as basic building blocks, and organizes them in such a way that a complicated task can be accomplished by coordinating the activities of many different agents. The foundation of this architecture is to classify agents into different function groups and manage each of them with a new, special agent called task/information manager (TIM). Individual agents and the TIM communicate directly, and two agents communicate with each other through the TIM.

A TIM has two main functions: (a) obtaining and maintaining the information of agents in the group, including their status and capabilities; (b) acting as a mediator between an agent requesting services and agents providing services to the request. If there is an available agent that can handle the request, the TIM simply acts as a matchmaker by routing the request to the capable agent. Such a request or task is called *primitive*. Otherwise, the task is called *non-primitive*. The TIM needs a plan of action to finish a non-primitive task. We use a database of predefined plans to choose a feasible plan for a given task. Given a service request, the TIM identifies all the applicable plans and chooses the most promising plan to execute. If the execution of the current plan fails, the next most promising plan is chosen for execution. This process continues until a plan succeeds, or all applicable plans are exhausted.

The first step in developing the multi-agent architecture is to organize the agents into groups. For each MIIIRO client, we have identified four groups: (1) Input group that includes Joystick, Head Tracker and Voice Agents; (2) Data group that has Data Management Agent and Plan Monitoring Agent; (3) Visual-Audio group that contains the visualization agents for Cockpit, Terrain, Auditory and Payload; and (4) Interface group that includes Interface Agent, Mission Planning Agent, Plan Validation Agent, Shared Control Agent, and Cooperative Work Agent. Further, we introduce a new type of agents, called Task/Information Managers (TIMs), to coordinate and manage these groups. Furthermore, the TIMs are connected according to their group functionalities. Figure 2 shows these four agent groups, their corresponding TIMs, and the connections among the TIMs. Notice that in an agent group, an agent does not communicate directly with other agents in the group, but only with its group TIMs.

# 3 Mission Planning

In MIIIRO, mission plans are represented by waypoints and targets using the Motion Guides and Task Lines concept [3]. The waypoints define the flight path that the UAV will fly. The targets define the locations and contain task commands to be executed at those locations. Target task commands include taking closeup images, launching weapons, operating payloads, etc. Waypoints may also contain waypoint task commands related to flight and navigation control.

The following sections describe how an operator uses the MIIIRO Mission Planning Agent, shown in Figure 4, to plan the different aspects of a UAV mission.

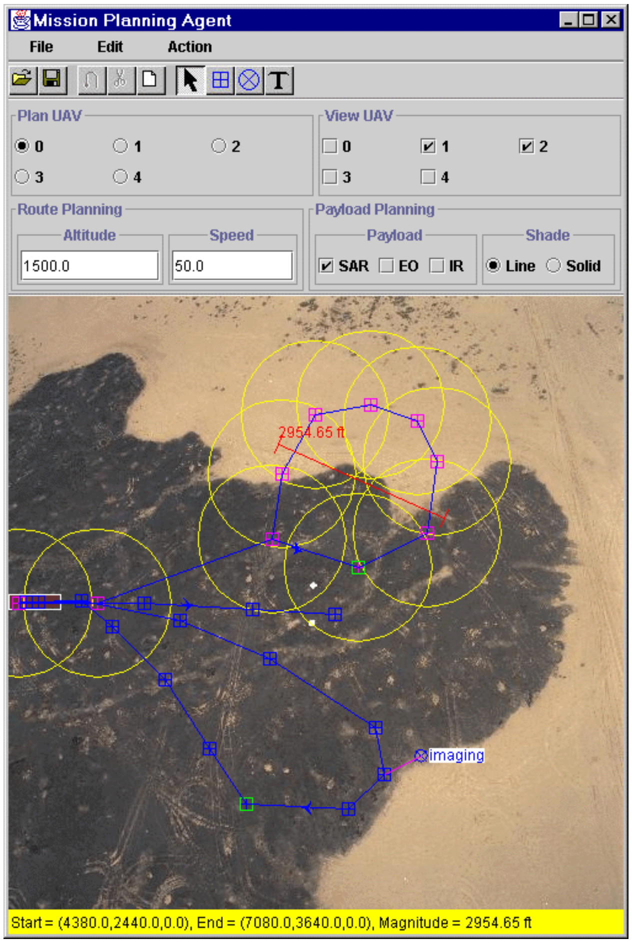


Figure 4: Mission Planning Agent

## 3.1 Navigation Planning

Navigation planning is used to specify the flight path of the UAV for a mission. The Mission Planning Agent allows the operator to visualize the mission area and enter waypoints at the locations directly on the terrain image. Each pixel in the terrain image corresponds to the actual location in the mission area. This relieves the operator from knowing the detailed coordinates and doing calculations.

Figure 4 shows the navigation plans of three typical mission scenarios. (1) *Suppression of Enemy Air Defense (SEAD)*: The upper plan in the figure shows a task plan for a SEAD mission in which the UAV flies to a designated area and then flies in a orbit pattern for enemy suppression. (2) *Intelligence, Surveillance and Reconnaissance (ISR)*: The lower plan shows a task plan for an ISR mission in which the UAV flies to mission area, performs reconnaissance, and returns. (3) *Heading Navigation*: The middle plan shows heading navigation. The operator can specify a few waypoints to bring the UAV to the desired direction and altitude. After the last waypoint is reached, the UAV will continue to fly in the same direction and altitude until the operator resumes control of the UAV.

## 3.2 Task Planning

Task planning is used to specify locations and the task commands to be executed at those locations. By clicking on a target using the mouse in the Command mode, the operator can open the Target Task Window to specify tasks to be performed at that target. The command window shows the location and altitude of the target. The window displays a list of predefined macros each of which consisting of one or more task commands. To specify task commands for the target, the operator can enter the text in the text area, or select a predefined macro to automatically enter its commands in the text area. Usually each task command is associated with a number of parameters. MIIIRO provides a Command Panel for the operator to select the parameters for each command.

## 3.3 Sensor Planning

Sensor planning is used to plan a flight path in order to meet the mission goal of a sensor. The flight path of a UAV is determined not only by the route reaching a certain location, but also the tasks to be performed in the mission. In a surveillance and reconnaissance mission, the UAV carries different sensors, including synthetic aperture radar (SAR), electro-optical (EO) imager, infrared (IR) imager, meteorological, radioactivity, chemical agent sensors and detectors, for gathering information in the mission area. The flight path of the UAV shall meet the task requirements of these payloads or sensors.

In MIIIRO, we implemented a simple mechanism within the Mission Planning Agent for sensor planning. The agent allows the operator to select the type of sensor, each with a different radius of operation. When a sensor is selected, its area of coverage is displayed at each waypoint, as shown in Figure 4. With this visual aid, the operator can plan the waypoints such that the whole mission area will be covered by the sensor's operation. The operator can show the circle of operation in either line or solid mode. It is easier to plan in line mode because the underlying terrain can be seen. On the other hand, it is easier to check whether the whole area is covered in solid mode.

### 3.4 Communication Planning

The communication planning is used to plan information distribution in order to meet the required performance of the communication channels. The MIIIRO stations continuously receive data from and send commands to the MIIIRO Server, over a local area network or the Internet. On the other side, the server also continuously receives data from and forwards commands to the UAVs, using a radio or satellite link. The communication bandwidths of these two channels affect the overall performance of controlling the UAVs greatly. If the satellite link cannot support the data rates of all the flying UAVs, or the network cannot support all the MIIIRO stations, commands and data will be delayed and even lost.

The communication planning is performed automatically by the MIIIRO system which makes use of its multi-agent architecture to coordinate the usage of communication bandwidth. The MIIIRO Server TIM monitors the bandwidth usage constantly. When the usage reaches a certain level, the server takes measures to reduce the data rates, without sacrificing the performance of control. There are several measures the server can take, depending upon the mission requirements:

1. Not to update the UAV flight and payload information for the MIIIRO stations not currently viewing the UAV.

2. Reduce the update rate of the UAV flight and payload information for the MIIIRO stations not currently controlling the UAV.

3. Reduce the update rate of the UAV flight information for the controlling station if it is currently under autonomous control.

4. Reduce the update rate of the UAV flight information for the controlling station according to the distance of the UAV from the operator's view point.

5. Not to update the payload status if the payload status is not currently viewed by the operator.

6. Use lossy image compression techniques to reduce the size of the images.

Most of these bandwidth reduction rules need the coordination between the server and the client agents. For example, only the Viewing Control Agent has the knowledge about what the operator is viewing. The server needs to consult the Viewing Control Agent of each station to plan its actions.

## 4 Situation Awareness

Situation awareness is imperative for the operator to effectively control the UAVs and payloads. MIIIRO has been designed to maximize the situation awareness by providing the operator with multiple views of the engaging area simultaneously, and a virtual environment to induce a sense of presence. A Viewing Control Agent is also developed to assist the operator in changing the view from one place to another. Figure 5 shows the 3D Virtual View developed with Java 3D, and the 2D Top View, Side View and Cockpit Instrument View developed with Java 2D.

### 4.1 Virtual View

MIIIRO uses the virtual window concept to provide the the virtual environment for inducing a sense of presence in the engaging area. Although the helmet-mounted display (HMD) is commonly used to provide virtual environment for situation awareness, in recent years, the virtual window approach, a high-resolution screen coupled with a head tracker that dynamically updates a 3D world according to the user's head position, has become popular as an alternative to the narrow field-of-view and low-resolution HMD [4, 5, 6]. Considering that a large amount of information needs to be presented and that there are many user interface elements such as menus, buttons, and lists, which are in 2D, we conclude that a setup with a large monitor or dual monitors of high resolution will better server the operator for visualizing mission situation.

The Virtual View is developed using the Java 3D API which is a standard extension of the Java 2 platform. Java 3D also provides a number of loaders for loading models specified in different file formats, including Virtual Reality Modeling Language (VRML). We modeled the Global Hawk UAV in VRML which is loaded into the Virtual View as a Java 3D object in the scene graph, with its behaviors programmed in Java 3D. This approach combines the best use of VRML and Java3D because VRML is an easy to use language to describe models while Java3D has the performance and versatility in interactions.

### 4.2 Top and Side Views

In addition to the Virtual View, MIIIRO also provides the Top View and Side View for the operator to visualize the situation from two orthogonal directions, as shown in the right side of Figure 5.

The Top View can best be used to view the locations of the UAVs in the engaging area. It also shows the flight

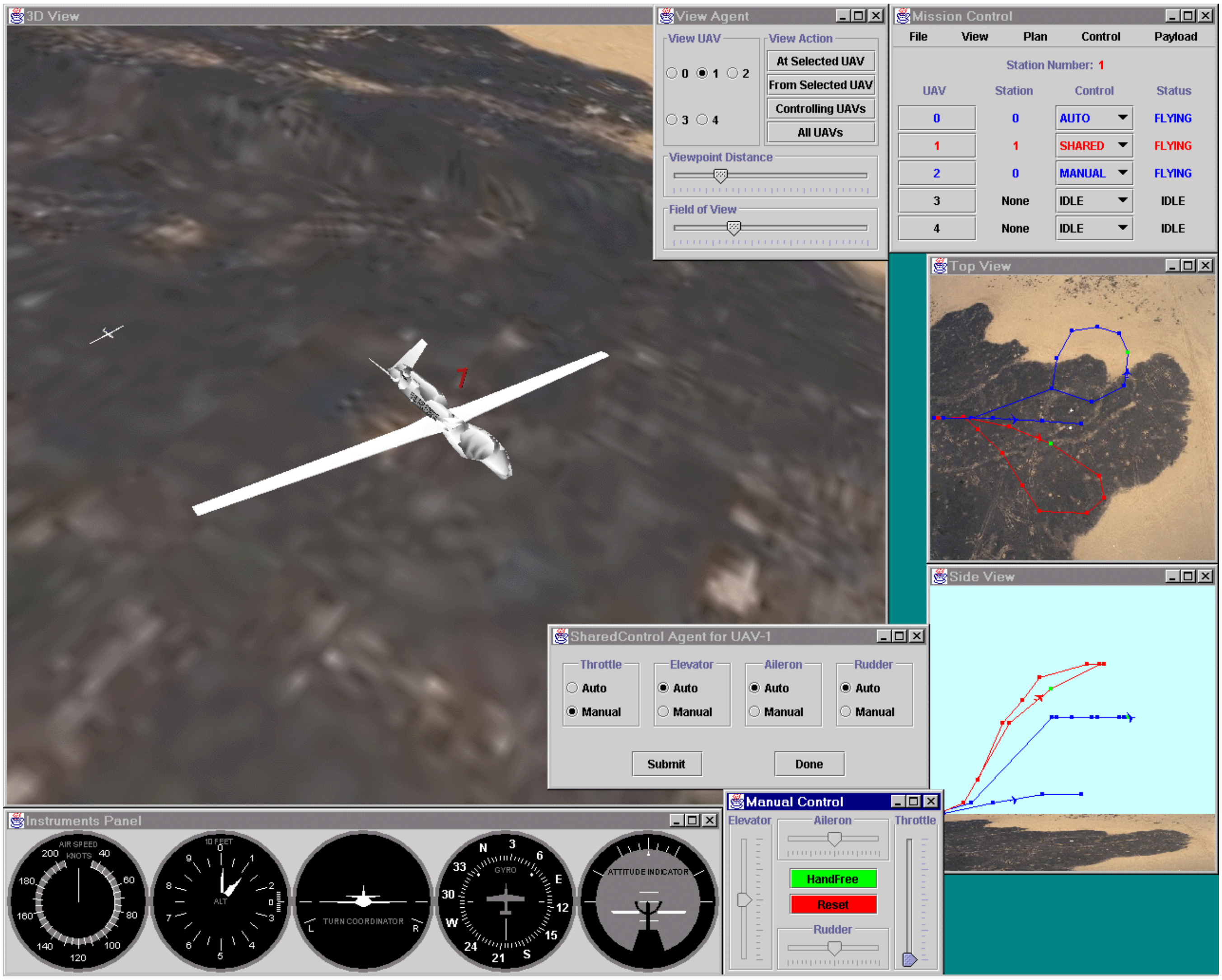


Figure 5: Snapshot of the MIIIRO Client

plans of the UAVs so that the operator can spot easily whether the UAVs are flying according to the plans, or whether a collision is imminent.

The Side View supplements the Top View in showing the altitudes of the UAVs. It is useful in spotting whether a UAV is flying too low relative to the terrain, or whether two nearby UAVs are flying at the same altitude such that potential collision can occur.

## 4.3 Cockpit Instrument View

The Cockpit Instrument View displays the data of the flight instruments pictorially similar to what a pilot usually sees inside a cockpit. Such a display can minimize the mental transformations the operator must apply to create useful information because the familiar cockpit display is already a part of a pilot's experience. The display includes the the air speed indicator, attitude indicator, altimeter, turn coordinator, heading indicator, and vertical speed indicator, as shown in the bottom of Figure 5.

## 4.4 Viewing Control

MIIIRO also provides a Viewing Control Agent for the operator to quickly and easily change the view from one part of the engaging area to another. It has four buttons to view the places where the operator is most likely to be interested:

1. *At Selected UAV*: The operator can select any one of his/her controlling UAVs in the Mission Control window. Using this button, the operator can view the selected UAV from outside, like viewing it from a spotting plane.

2. *From Selected UAV*: This button allows the operator to view the world from the cockpit of the selected UAV.

3. *Controlling UAVs*: This button allows the operator to view all the UAVs currently under his/her control.

4. *All UAVs*: This button allows the operator to view all the UAVs, including those controlled by other operators, which are currently flying.

In addition to the buttons, two sliders are provided for controlling the field of view and distance from the UAVs. The operator can also use the head tracker to change the viewing direction.

# 5 UAV Flight Control

For long range, high endurance UAVs, it is infeasible to demand the operator to teleoperate the vehicle continuously or to pre-program a complete flight and task plan for the whole mission. The operator will need to plan and control the UAVs in real-time based on information received from the flight instruments, navigation system, terrain images, and various sensors. Therefore, MIIIRO provides three control modes for operating the UAVs: manual, autonomous, and shared control, which will meet the needs of real-time control. The MIIIRO system also ensures that the switching between these three modes is seamless, i.e., the UAV will not lose control during the switching and no uncommanded motion will occur. The operator uses the Mission Control and Status window to select one of the three available control modes, as shown in the top right window in Figure 5.

## 5.1 *Manual Control*

In manual control, the operator uses the inputs from the joystick to teleoperate the selected UAV. When MIIIRO is run on computers which do not have a joystick input device, we also developed the manual controller with sliders for throttle, elevator, aileron, and rudder control, as shown in bottom right window of Figure 5. The inputs from the joystick or manual controller are then sent to the corresponding UAV or UAV simulator for controlling its speed, heading, and turning motion.

## 5.2 *Autonomous Control*

In autonomous control, the UAV flies according to a submitted flight plan and executes task commands at specified locations. The flight plan is specified with waypoints indicating the locations to which the UAV should fly. Tasks are specified with targets indicating the locations at which task commands are executed. To ensure seamless switching from manual mode to autonomous mode, the UAV simulator will first determine the waypoint to which it should fly, depending on its current heading and the distances from the waypoints in the task plan. After the waypoint is identified, it will gradually change its course to fly to that waypoint, and then to the next waypoint according to the task plan. The waypoint the UAV is flying to is indicated in the Mission Planning Agent, Top View and Side View so that the operator can always be aware of where the UAV is heading.

## 5.3 *Shared Control*

Shared control is the mixing of inputs from the operator and the autonomous system while flying a UAV. Shared control is useful in many situations, such as (1) maneuvering to avoid an attack, which requires adding perturbation to the current flight path, and (2) flying the UAV at a low altitude temporarily in order to capture closeup images, without changing its course.

The operator first specifies the axes to be controlled autonomously according to the current flight plan, and the axes to be controlled manually. After submitting the selections to the UAV, the operator can then use the joystick to add inputs to the autonomous flight control. Only the axes which are selected for manual control will be affected by the manual inputs.

# 6 Multi-User Collaboration

MIIIRO supports multiple users to plan and control the UAVs collaboratively. Collaboration provides a number of benefits: (1) It can reduce crew size. Knowing that another operator can take over control, an operator can control two or more UAVs at the same time. In the situation where two UAVs require full attention at the same time, the operator can request another operator to take over the control of one of the UAVs. (2) A commander can examine the task plans and observe the situation of the mission. The commander can make changes to existing plans or submit new task plans. (3) The operators can alert and help each other since everyone can visualize the complete situation and observe the UAVs controlled by other operators.

## 6.1 Control Sharing

The control of multiple UAVs can be shared by two or more operators. When an operator is involved in some demanding situation, he/she can ask a colleague to take over control of one or more UAVs currently under his/her control. An operator can also initiate an un-invited control takeover when workload is relatively light and a colleague is in a demanding situation.

The operator uses the Mission Control window, the top left window in Figure 5, to request takeover control of a

UAV. The window has a column of buttons each of which is labeled with the UAV's Id. Next to the UAV button is the Id number of the MIIIRO station controlling that UAV. If an operator wants to control a particular UAV, say UAV-2, he/she can initiate the control by pushing the UAV-2 button. The request will be sent to the MIIIRO Server which will respond according to the situation as follows: If no station is currently controlling that UAV, the server will grant the control immediately and informs all stations that the UAV is now under the control of the requesting station. If the UAV is under the control by another station, the server will engage the two operators in a dialog to ensure that both parties agree on the control takeover.

### 6.2 Plan Sharing

MIIIRO allows each operator to view the task plans associated with the UAVs. This is important for commanders to oversee the overall mission and for operators to develop new task plans.

An operator uses MIIIRO's planner to enter waypoints and targets over a terrain map to plan for a UAV mission, as described in Section 3.1. The planning is conducted locally in order to reduce communication and minimize distraction to other operators. But once the task plan is completed and saved, it will be propagated to all the MIIIRO stations.

### 6.3 Situation Sharing

MIIIRO allows system states to be shared by all the operators. The server updates the Mission Control window with information about the controlling station, control mode, and overall status of each UAV. The state (speed, location and orientation) of each UAV is also updated in the Virtual View, Top View, Side View and Cockpit Instrument View windows. This allows the commanders to observe the mission execution. Operators can also help each other and take over another operator's UAV by knowing the complete situation of the mission.

## 7 Conclusion

This paper described a scalable, expandable and platform-independent operator interface system for controlling future long range, high endurance unmanned aerial vehicles. MIIIRO, developed based on open Internet technologies including the Java 2 platform, Java 3D, VRML, RMI, and CORBA, enables collaborative planning and coordinated control of UAVs over a network for achieving a common mission goal.

Our work now is focused on the Interface Agent which is envisioned to be a personal assistant cooperating with the operator in planning and control the UAVs [7]. The Interface Agent tries to understand the goal and intention of the operator based on the head motion and other inputs, the task context and current situation. When the need of the operator is identified, the Interface Agent invokes the services of other agents to satisfy the operator's desire. When the operator needs to watch for a number of concurrent activities, the Interface Agent coordinates the visualization agents to determine where on the display and what information should be presented to the operator.

## Acknowledgments

This work has been supported by the Air Force Research Laboratory Crew Systems Interface Division under the Small Business Innovative Research (SBIR) contract F41624-98-C-6006. The authors wish to express our appreciation of the interest, support and guidance of Dr. Robert Eggleston, Project Manager of this research.

## References

[1] M. Benda, "Internet architecture: Its evolution from an industry perspective," *IEEE Internet Computing*, vol. 2, pp. 32–35, Mar. 1998.

[2] W. Zhang, R. Patil, and W. Shen, "An information mediator network for tasks in dynamic environments," in *AAAI-96 Spring Symposium on Artificial Intelligence in Medicine*, (Stanford, CA), pp. 174–178, Mar. 1996.

[3] P. G. Backes, S. F. Peters, L. Phan, and K. S. Tso, "Task lines and motion guides," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 1, (Minneapolis, MN), pp. 50–57, Apr. 1996.

[4] M. Deering, "High resolution virtual reality," *Computer Graphics*, vol. 26, no. 2, pp. 195–202, 1992.

[5] K. W. Arthur, K. S. Booth, and C. Ware, "Evaluating 3D task performance for fish tank virtual worlds," *ACM Transactions on Information Systems*, vol. 11, no. 3, pp. 239–265, 1993.

[6] G. Tharp, S. Hayati, and L. Phan, "Virtual window telepresence system for telerobotic inspection," in *Proceedings SPIE Telemanipulator and Telepresence Technologies Conference*, (Boston, MA), Oct. 1994.

[7] R. G. Eggleston, "Adaptive interfaces as an approach to human-machine cooperation," in *Proceedings of the 7th International Conference on Human-Computer Interaction*, (San Francisco, CA), Aug. 1997.