HUMAN INTERACTION WITH SOFTWARE AGENTS
PHASE I FINAL REPORT
SBIR TOPIC SB992-042

Contract DAAH01-00-C-R054

Prepared For:

Defense Advanced Research Projects Agency
Arlington, VA

Daniel H. Wagner Associates, Inc.
2 Eaton Street, Suite 500
Hampton, VA 23669

Submitted by:

Dr. Joseph H. Discenza
Mr. James T. Eanes
July 6, 2000

REPORT DOCUMENTATION PAGE

Pulic reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching exisiting data sources, gathering and maintaining the data needed, and completing and reveiwing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and the Office of Management and Budget, paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE 6JUL00	3. REPORT TYPE AND DATES COVERED Final Report 29OCT99 – 07JUL00

4. TITLE AND SUBTITLE

Human Interaction with Software Agents

5. FUNDING NUMBERS

Contract #
DAAH01-00-C-R054

6. AUTHOR(S)

Dr. Joseph H. Discenza
Mr. James T. Eanes

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Daniel H. Wagner Associates, Inc.
2 Eaton Stl, Suite 500
Hampton, VA 23669

8. PERFORMING ORGANIZATION REPORT NUMBER

Case 7570

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

US Army Aviation and Missile Command
ATTN: AMSAM-RD-WS-DP-SB
Bldg. 7804, Room 205
Redstone Arsenal, AL 35898

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT (see Section 5.3b of this solicitation)

Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

Report developed under SBIR contract for topic SB992-042. Software agents are small computer programs that perform specific operations for users based on those users' wishes or needs. The benefit of an agent is that it can be programmed to provide just the information needed by that user, at the right time and place for that information to be useful. Agents can save users enormous amounts of time by avoiding the need for developing complicated searches and extract scripts every time new information is required. This report details the results of Phase I SBIR research for DARPA on the concept of Agent Wizards, or programs that can automatically create agents for users without the need for detailed programming or significant computer knowledge on the user's part.

14. SUBJECT TERMS SBIR Report, Software Agents, Agent Wizard, Environment, METOC, Distributed Computing, DEI, CORBA, XML, CoABS, DAML

15. NUMBER OF PAGES
55

16. PRICE CODE

17. SECURITY CLASSIFICTION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UNLIMITED

REF E

EXECUTIVE SUMMARY

Software agents are computer programs that perform specific operations for users based on those users' wishes or needs. The benefit of an agent is that it can be programmed to provide just the information needed by that user, at the right time and place for that information to be useful. Agents can save users enormous amounts of time by avoiding the need for developing complicated searches and extract scripts every time new information is required. In the context of military mission planning, an agent being developed by Daniel H. Wagner Associates, Inc. called "METPLAN" will provide a cost effective means for developers of mission planning decision aids to include environmental information. This report details the results of a Phase I SBIR research for DARPA on the concept of "Agent Wizards," or programs that can automatically create agents for users without the need for detailed programming or significant computer knowledge on the user's part. In the Phase I we created an architecture and design for an Agent Wizard together with a prototype user interface to demonstrate the concepts. We also developed ideas for commercial versions of the Agent Wizard and developed a consumer weather agent that works within a browser to provide travel-related weather information to an online travel planner. We have obtained agreements for Phase II and Phase III research cooperation and technical support and are continuing to attempt to secure funding commitments for Phase III. We have at least one potential industrial customer for a commercial version of the weather agent. Successful development of a complete Agent Wizard prototype in Phase II will have a significant benefit for DoD programs, developers, and users in the mission planning community.

TABLE OF CONTENTS

1. Overview of Phase I

1.1 Project Goals

The goal of this SBIR project is to provide a means by which users can easily create software agents for performing data discovery and retrieval. We accomplish this goal by creating a "wizard" tool for creating software agents and by providing a framework for future work on developing software agents in the military information space.

We have designed a tool, which we call the Agent Wizard, for creating and controlling intelligent software agents. Agents, in general, are persistent software programs capable of providing specialized services over time. The agents created by the Agent Wizard will be responsible for locating and retrieving data from information resources on heterogeneous networked systems according to a set of rules and preferences created by the user.

The Agent Wizard contains a graphical user interface (GUI) to allow the user to define the rules and preferences for data retrieval tasks. When the user has defined a task, the Agent Wizard creates and dispatches an agent to perform that task. The Agent Wizard GUI provides the user with control of all agents and the data they retrieve. When the agent has completed its task, the Agent Wizard alerts the user that the agent has results to display. These results will be URLs of discovered web pages, textual descriptions of web page metadata, and textual descriptions of database records. The user can then download the data associated with a given link. He can also refine the search based on those results that best match his intent and dispatch the agent to continue searching.

When an agent is created and dispatched by the Agent Wizard, that agent attempts to locate and retrieve information from the given information space. In the case of the military, the information space will include three tiers: the Global Command and Control System (GCCS), the Secure Internet Protocol Routing Network (SIPRNET), and the Internet. On GCCS, the agent will be responsible for searching proprietary databases through their published interfaces and for searching Intranet web pages. On SIPRNET, the agent will search web pages that are under military control. And on the Internet, the agent will search web pages under the separate control of individual entities.

For commercial agents, the information space will consist of two tiers: the company's Local Area Network (LAN) or Wide Area Network (WAN), and the Internet. Again, content on the Internet is separately controlled by all of its members, but the company-wide information (or that shared between cooperating companies) will be stored in proprietary databases with published interfaces.

1.2 Phase I Accomplishments

In Phase I, we identified the markets for software agents, we designed the Agent Wizard architecture, we developed a demonstration of an Agent Wizard, we studied Agent language possibilities, and we designed the distributed agent architecture for Phase II. In the Phase I, we focused strictly on the domain of environmental data, because that is the area of expertise that we have developed from our Navy Phase I, II, and III activities.

We examined both the defense and commercial markets for substantially similar applications in the same (weather) domain. We identified a substantial set of defense clients and are already pursuing many of these clients in connection with our Navy Phase III work. We also identified a list of commercial clients whose current product lines are consistent with the use of agents for weather data. We also filed for an interim patent for the commercial Weather Agent.

1

In designing the Agent Wizard, we utilized the concepts of CORBA for universality of data access and have included use of the JAVA language for universal hosting, especially for development of the GUI. The Agent Wizard will be heavily dependent on the method of managing the metadata and will therefore need to have several versions, for DEI, CORBA, XML, and other methods.

1.3   Methods for Accessing Data

In order for the Agent Wizard to generate software agents automatically, we must first be able to create software agents. This, in turn, requires knowledge of the information space. This knowledge is acquired statically, before the agent is built, and dynamically, through resource discovery. A resource is any object that provides information. This object can be a database, a web page, a program, or another agent.

Static resource descriptions, such as APIs, provide programmers of static code with valuable insight into information resources at the time of software development. In contrast, dynamic resource descriptions allow software to find data and functionality that were not known at compile time. Much work is being done to allow agents to dynamically discover new resources on the networks on which they live. Below, we briefly review two major approaches in this area.

One method for discovering new resources is to actively look for them. A Web Crawler is an agent whose task is to search web pages on the Internet (or any network) for new information to return to its master for display or processing. The agent's master can be the end-user, another program, or the agent itself. Web Crawlers typically execute simple search algorithms, traversing Universal Resource Locators (URLs) and collecting metadata from each. That metadata is then presented to the user for resource collection or further search refinement.

Some web crawlers try to parse the Hypertext Markup Language (HTML) that makes up the web pages they find. This approach has yet to prove effective. The problem is that an HTML page is made of up several disparate interspersed components, typically HTML tags, programming-language scripts, graphics, and text. Tags and scripts (e.g., JavaScript) provide information about how to display and manipulate data and graphics on the page, but nothing about the data itself. In addition, current Web Crawlers have no way of interpreting graphics and a very limited ability to understand natural language text (e.g., English),

Other Web Crawlers look for metadata schemas in the HTML pages they find. Metadata is information about data, and a schema is a method of encapsulating metadata. For example, a Web Crawler may look for <META> tags in an HTML page to determine whether those tags contain fields that match the schema with which the Web Crawler is familiar.

Common schemas include Dublin Core [2] and the Global Information Locator Service (GILS) [3]. Both schemas claim to encapsulate any electronic resource.

The following example shows HTML <META> tags containing metadata recognized by an intelligent agent that knows the Dublin Core schema. When the agent locates HTML with <META> tags, it checks the information within those tags to determine whether it fits one of its known schemas. If it does, the agent collects the metadata and returns it to the master for further instructions.

2

```
<META NAME="DC.Title" LANG="en" CONTENT="Zen Buddhism">

<META NAME="DC.Creator" LANG="en" CONTENT="John Smith">

<META NAME="DC.Description" LANG="en" CONTENT="Historical
perspectives on Zen Buddhism and its impact on Japanese
culture.">
```

The Dublin Core metadata shown above was created using a metadata editor called Reggie [4]. Reggie provides a user interface for entering schema data elements. The metadata it creates is in the user's choice of HTML 3.2, HTML 4.0, Resource Description Framework (RDF), or Abbreviated RDF.

The main problem with the schema methodology today is that, even though there are tools like Reggie for easily creating metadata, most web site builders are not using metadata in their HTML. The only <META> tag on most web pages contains the name of the HTML editor used to create the page (e.g., FrontPage, Mozilla). Therefore, a Web Crawler looking for useful information is currently limited to a small subset of the data that is actually available.

An example of a Web Crawler that uses metadata effectively to search a restricted set of web pages is a Java-based agent called HotMeta. HotMeta was created by Australia's Distributed Systems Technology Centre [5] to search known databases for information based on the Dublin Core schema. Figure 1 shows HotMeta's search screen, which allows the user to provide criteria for each element in the schema. Here, the user is choosing to look for "business" in the Title element.

The interface provides the option of matching the given words exactly or "stemming," which is a process of applying rules to discovered text to determine whether it matches the search criteria. For example, if the Title element is set to "business" and stemming is turned on, then the agent may return Title fields containing the word "businesses" as well as "business." The agent's creator determines the rules for stemming, so this agent may or may not return Title fields containing "businessmen." Turning off stemming allows the user to reduce search time by forcing the agent to disregard unwanted information.

Figure 1. HotMeta Search Screen

Figure 2 shows the Search Results screen, allowing the user to refine his search, view the web page referenced by the metadata, or view the entire metadata record. The top part of the screen shows that this was a Title search of those web pages using the Dublin Core (DC) schema. The Refinements section provides canned topic headings containing the search string.

Figure 2. HotMeta Search Result

HotMeta provides an easy-to-use interface with powerful search capabilities. However, like all other metadata-based web crawlers, it is limited to those web pages that contain metadata of its schema. This is shown in the example search above, where the HotMeta user receives only Australian government sites that utilize Dublin Core metadata.

Another method of allowing agents to perform resource discovery is to supply a distributed object management system. Objects in this context are databases, agents, display programs, word processors, etc. This method requires an object server to which every object publishes itself. This allows objects to query the server for access to other objects. There must also be a language by which all objects communicate with the object server.

For example, if we have an agent object interested in finding information about teachers, the agent will make a request of the object server to retrieve all information about teachers. Say we have a teacher database that has registered itself with the object server. The server will make the request of the database and return the results to the agent.

There are currently two major object management systems available: the Common Object Request Broker Architecture (CORBA) and the Distributed Component Object Model (DCOM).

CORBA was developed by the Object Management Group (OMG) to "allow intelligent components to discover each other and interoperate on an object bus" [6]. CORBA's object server is called the Object Request Broker (ORB). The ORB maintains published object information to allow objects to discover each other at run time and invoke each other's methods. The language by which the objects communicate with the server is the Interface

Definition Language (IDL). IDL is a superset of C++, with additional keywords specific to distributed applications.

The Microsoft version of the object management system is DCOM, the distributed version of their Component Object Model (COM). Microsoft developed DCOM to act as the backbone for all of its distributed computing ventures. COM is the basis for Object Linking and Embedding (OLE), which is now known as ActiveX. In this architecture, objects on the same machine communicate via the COM runtime library, and objects on different machines communicate via the DCOM Network Protocol, which connects the COM runtime libraries on the different machines. The DCOM Network Protocol is based on the Distributed Computing Environment (DCE) Remote Procedure Call (RPC).

Both object management systems provide object communication and resource discovery. Both systems have fostered the development of tools for creating and maintaining their respective objects. And, both systems provide a high-level protocol (IDL or DCE RPC) that can be wrapped around legacy systems to allow them to communicate with other objects.

We have discussed two methods by which an agent can acquire knowledge about the information spaces it is assigned to search: crawling the space, and being part of a managed space. Now, we will discuss what we already know about the information spaces so that we can assist the agent in completing its tasks in the most timely and effective manner possible.

The military information space consists of three tiers: GCCS, SIPRNET, and Internet. Because of the large number of legacy systems, the access to many existing databases is through single-purpose pipes. Classically, these pipes provide one group of people with one type of information. On the positive side, these databases usually have well-defined application programming interfaces (APIs) and many are becoming controlled by a military standard, the Defense Information Infrastructure Common Operating Environment (DII-COE). There are also dedicated personnel, often subject matter experts, associated with these databases. Web pages are also usually under strict control and have personnel dedicated to their monitoring and maintenance.

Until the military information space is controlled by an object management system such as the DEI system described in Section 3.2.4.1, the databases found on military networks will have this "stovepipe" architecture. If there were an object management system, say CORBA, on the network that an agent is assigned to search, then the agent would simply make his query to the object manager to pass on to the proper object, but without this type of object management, the agent must be familiar with the target databases' APIs in order to perform queries and retrieve the data from the database. This requires prior knowledge, therefore, limits resource discovery to those resources contained within known repositories.

HotOil is an example of this type of middleware [7]. HotOil knows how to query repositories using Hypertext Transfer Protocol (HTTP) and Z39.50, which is the ANSI standard for Internet information searching. When HotOil queries a known repository, it converts the results into Dublin Core metadata, merges the results and removes duplicates, and then displays the results to the user. This is very effective for those resources that use HTTP and Z39.50. Unfortunately, that does not include many proprietary database protocols.

Military web pages are also not currently maintained with resource discovery in mind. That is, they do not make use of metadata. The solution for this problem, though, is much simpler than introducing an object management system. What is needed here is a standard schema to be included in all HTML on military web sites. This would provide intelligent agents with the metadata they need to perform successful search of the military web space.

The commercial information space has similar issues with "stovepipe" systems and metadata-lacking web pages, but on a different scale. The solution for the first tier of the commercial information space is the same as for the military. Within a company's LAN or WAN, an object management system and a standard schema could be used to allow an agent to find the information needed by its master. However, the lack of control and standards among companies limits the possibilities for improving the Internet tier of the combined commercial and military information space. A standard schema, or at least a small subset of the available schemas, would have to be adopted by companies all over the world in order to provide metadata recognizable by software agents. This is the goal of many factions of the growing metadata community, even if they don't agree on which schema(s) should be standard. For example, the Parallel Understanding Systems Group at the University of Maryland proposes an HTML extension called Simple HTML Ontology Extensions (SHOE), which they propose could someday be as ubiquitous as HTML [8]. DARPA is currently launching a program to develop the DARPA Agent Markup Language with the goal of providing for agent communication between schemas and ontologies, so that one standard is not required [9].

2. The Market for Weather-Oriented Software Agents

The difficulty in finding and retrieving useful and accurate information quickly remains a serious problem. Though the latter part of the twentieth century has been labeled the Information Age due to the advent of mass communications technologies, it is still difficult to glean useful information from the increasingly large and complex network of data sources. The Internet has provided a gateway through which homes, businesses, and military installations can connect with each other as never before to share and sell all manner of data. However, finding the right data from the right place at the right time is often difficult, if not impossible.

2.1 Military Markets

The military's evolution to network-centric warfare has promised information superiority in the battlespace through the integration of information-rich systems on local and global networks. So far, this approach has provided effective data warehousing. Whether in a munitions database on a LAN, a weather database accessed over the Internet, or a secure web page providing details for an upcoming mission, remote data abound in this new information space.

However, the current dependence on disjointed legacy systems limits the amount of useful information provided to the end-user from the available data. The military has developed the Global Command and Control System (GCCS) to allow heterogeneous systems to share specific types of information (tracks, overlays, ASCII messages), but the protocols are limited to specific tactical data, and the retrieval methods are generally dependent on the databases involved. Other forms of information are confined to point-to-point pipelines between cooperative systems. Also, there is little or no communication with outside systems during joint or international operations.

This problem has prompted the Air Force Scientific Advisory Board (SAB) to suggest new methods of information management in what they call the Battlespace InfoSphere (BI) [1]. The SAB reports 11 findings concerning battlespace information management:

1. Combat information requires management
2. A staff function is required to manage information
3. Information validity is achieved through authenticating inputs and tracking information pedigree

4. Selected information needs constant updating
5. Data must be organized by referencing and cataloging
6. Data must be assembled into useful information
7. Information must be presented at the user's desired level of knowledge
8. Subscription or search finds the right information to meet user needs
9. Objects can be published for common sharing
10. The Battlespace Infosphere (BI) creates a common operating picture (COP)
11. Human control, with rule-based information decisions, is required to achieve rapid and accurate decision cycles that provide information superiority

Based on these findings, the SAB suggests a change of paradigm in the information management process to reduce the current "data overload and information starvation." Instead of providing a data "pull," where data needed by one system is provided by one or more servers in a format defined by the server, they suggest a "use-driven" methodology. This new approach would provide each system with an intelligent agent capable of accessing all available data in the battlespace to provide information based on the requirements defined by that system's end-users. The Battlespace InfoSphere consists of common objects (e.g., data, products, imagery), manipulation functions (e.g., publish, subscribe, transform), and representation components for end-user interaction.

The SAB gave the name "Fuselets" to those intelligent agents responsible for object publication, subscription, and processing. Figure 3 shows the overall design of the Battlespace InfoSphere as given by the SAB. Figure 4 shows the SAB's representation of the role of the Fuselet as an intelligent agent.

Wagner Associates has developed a Fuselet for providing meteorological and oceanographic (METOC) data to Joint Standoff Weapon (JSOW) mission planners. In a Navy Phase II SBIR project, we developed an agent called METPLAN that pulls data from the Tactical Environmental Data Server (TEDS) to provide the METOC data required to understand the environment during the missile's flight and at the target. The METPLAN GUI allows the user to set preferences for locating, retrieving, and displaying METOC data, products, and imagery. Based on these preferences, METPLAN connects to TEDS, queries for the correct data sets, and returns the requested data in the correct formats at the requested times. Figure 5 shows the Data Retrieval window from METPLAN connected to our prototype JSOW MPM.
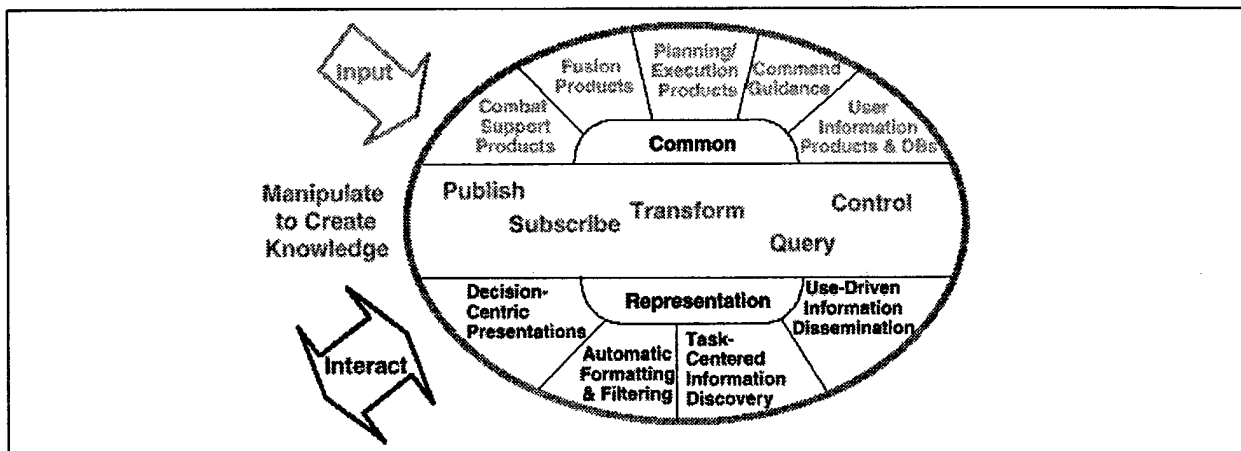

Figure 3.   Components of the Battlespace InfoSphere

7
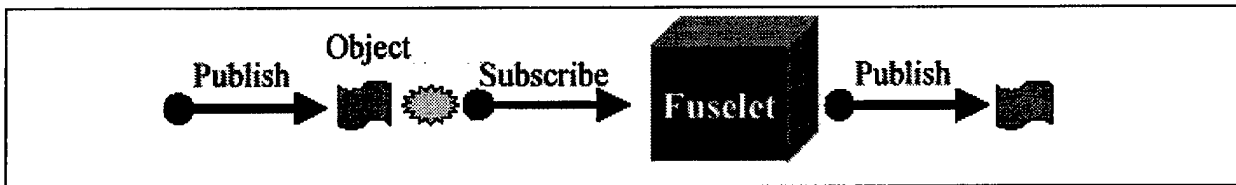
Figure 4.   Information Flows Using Publish, Subscribe and Fuselet Processing
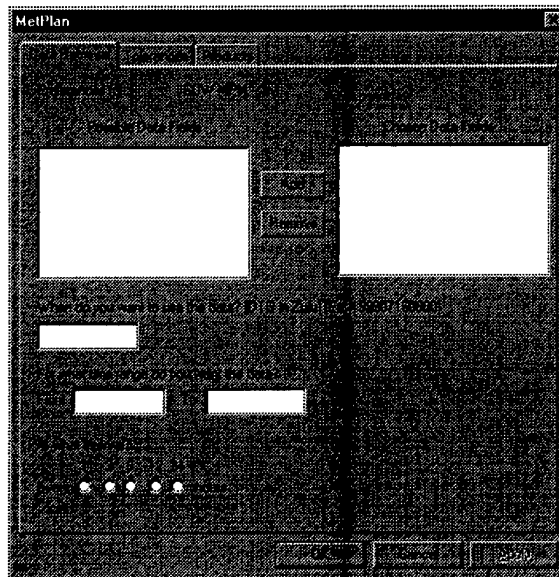
Figure 5.   Wagner Prototype JSOW MPM Data Retrieval Window for Accessing METPLAN's

When METPLAN has retrieved the requested data, it integrates that data into the prototype JSOW Mission Planning Module (MPM). Part of our research for this project was to capture end-user intent to ensure that the mission planners get the data they want in the manner they want. To accomplish this, we met with as many mission planners (F/A-18 pilots) as possible and compiled their suggestions into a list of functional elements to incorporate into our JSOW MPM. We then implemented those functional elements, which include point-and-click data retrieval, product-route overlays, and route coloring based on user-defined tolerances. A red/yellow/green stoplight display of the mission route allows the mission planner to see the effects of the weather without having to interpret METOC-oriented charts. Figure 6 shows the point-and-click data retrieval. This has been, by far, the most anticipated feature of the system, since the popup menu provides the planner with a cross-section of all available information for the chosen point.

8

Figure 6. Point-and-Click Data Retrieval at the JSOW Target

The choice to retrieve METOC data from TEDS was made for us. TEDS is the environmental server planned for the Joint Mission Planning System (JMPS), and it is the only METOC database constantly being updated w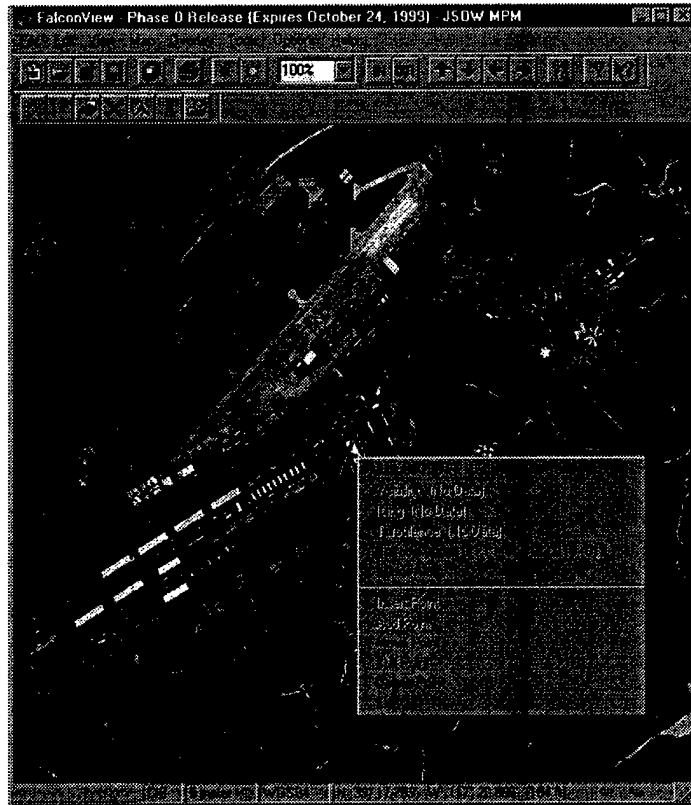ith data from the Coupled Oceanographic Atmospheric Mesoscale Prediction System (COAMPS). COAMPS provides the highest-resolution METOC data available (down to 3km), which makes it perfect for JSOW missions, which do not exceed 100 miles. In the global weather models, such as the Naval Operational Global Atmospheric Prediction System (NOGAPS), such a small area of interest could rest entirely between grid points.

The retrieval of data from TEDS is through the Application Program Interface (API) to the Grid Field Database (MDGRID). This API provides functions for retrieval of data catalogs and specific grid fields. METPLAN successfully utilizes this API to provide the JSOW MPM with the data fields requested by the mission planner at the times he has requested and in the formats required by the JSOW MPM.

METPLAN, then, is a Fuselet like those defined in the SAB report. It provides a legacy system with data from a remote system with some intelligence concerning the data retrieval. However, METPLAN is limited to one field of information, namely METOC, and one remote database, namely TEDS. So, though METPLAN provides much needed environmental data to JSOW mission planners, it cannot provide them with information from any other area.

What the end-user needs, in general, is a method for creating an intelligent agent for gathering information from the entire information space. However, end-users often do not have the resources for writing software to retrieve the information they need. Therefore, what is needed is a tool with which the user can create, dispatch, and control intelligent software agents for retrieving data from the information space.

9

Our goal in Phase I was to build upon our work with METPLAN to develop concepts and a demonstration of an Agent Wizard to create adaptable intelligent agents and to establish the framework necessary to meet military and commercial information management requirements. The principal customers are:

- METOC Officers -- create agents for METOC data collection/review to provide their customers (warfighters) with reports/presentations of information gleaned from disparate sources on the network.

- Military Mission Planners (pilots, searchers, etc.) -- create agents to provide timely information from disparate sources for mission planning. Already have users in JMPS (JSOW, SLAM-ER, Tomahawk, etc.) and ASUWTDA.

- Modeling and Simulation Community -- create agents for environmental data collection to run scenarios (e.g. Environmental Scenario Generator is rolling their own METOC retriever to pull data from Master Environmental Library (MEL)).

- Military Data Base Managers – provide users with easily used tools to create agents to access their databases.

2.2 Commercial Markets

Many commercial and private interests have a strong need for weather information and data as part of routine consumer and business activities. The most obvious of these involves traveling, whether deciding on a vacation destination, selecting an airline route to avoid bad weather, or simply determining the dress required on a coming trip. The need for weather information is more specific when the user is a pilot planning a flight or a boat owner deciding whether to venture out to sea. Potential customers are:

- Private Pilots -- create agents to keep up with changing data providers (and changing formats and systems among existing providers) of environmental and aeronautical information.

- Commercial Airlines -- create agents to gather and coordinate information for dissemination to pilots.

- General Web Users -- create agents (e.g. WeatherDog, patent filed) to provide context-sensitive and automatically updated information from web resources (weather, stocks, real estate, etc.).

- National Park Service -- create agents to gather environmental information for fighting forest fires (and determining when to set planned, controlled fires).

- Financial Institutions -- create agents to gather/monitor/alert of changes in monetary information from network resources.

- Large Corporations -- create agents to gather/monitor information within their corporate LAN; provides for better/faster access to information within the corporation and reduces network bottlenecking for overused resources.

Many sources for weather information already exist on the world wide web. The preeminent source is weather.com, a unit of The Weather Channel.

The Weather Underground, Inc., agriculturalweather.com for agricultural interests, CNN Weather, EarthWatch® Weather On Demand, Yahoo!, NiceWeather.com, Intellicast, and many others all have web sites that allow users to select locations and obtain short-term forecasts. The WeatherPlanner site claims to be able to predict local conditions up to a year in advance.

Webbers Communications provides Weatherbyemail.com and Weatherbypager.com which deliver these forecasts automatically on a subscription basis; currently these services are free.

All of these private services depend on the National Weather Service (NWS) for collection of data and creation and distribution of graphical products from weather satellites. They utilize their own web pages to provide the services. Weather Underground also provides a free banner that other web sites can use to fetch and display real-time weather conditions for a specific location (e.g., http://home.earthlink.net/geo/DefaultBannerPromo/US/VA/Poquoson .html).

2.3 Competition

There are some agent-like programs that are offered in the marketplace to allow a user to obtain weather information without having to use the provider web sites.

WeatherTracker by Alladin Systems. This MAC or WIN program connects to a weather server (The Weather Underground) and displays current conditions on the screen. The user can also receive special products such as local forecasts by city, marine forecasts, and climatology. The program requires the user to set up a set of cities to monitor and each appears all the time on the user's desktop.
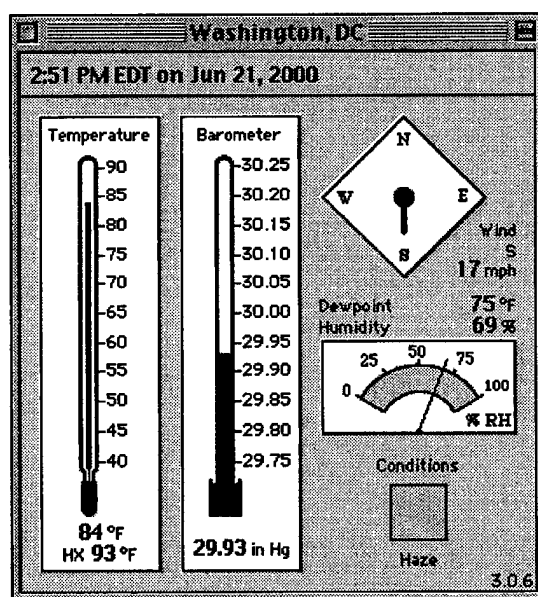

Figure 7. WeatherTracker Software Agent

Weather Watch is an online 4-window browser designed to display weather maps(radar and satellite) for a selected state or region. It provides precipitation, storm movement, and severity. It also displays information on hurricanes, their tracks, and weather advisories. Any of the four smaller windows can be brought up in full screen.

11

3. Phase I Technical Objectives and Specific Accomplishments

3.1 Technical Objectives

The goal of our Phase I work was to design the Agent Wizard, a tool for creating intelligent software agents, and to explain its expected benefits to military and commercial information management.

Objective 1. Select a subset of the military information space to search and a schema to encapsulate that information. Clearly, in Phase I we could not develop an Agent Wizard that generates and controls agents for arbitrary databases and knowledge domains. To demonstrate the effectiveness of our approach, we needed a bounded domain in which to work.

Objective 2. Develop and test a generic agent capable of searching the chosen information space using the given schema.

Objective 3. Develop and test the Agent Wizard for creating and controlling multiple agents.

Objective 4. Determine the most suitable schema(s) and object management architecture(s) for future work.

3.2 Specific Accomplishments

3.2.1 Objective 1: Environmental Data for Mission Planning

We selected an obvious subset based on our experience and current efforts with the U.S. Navy mission planning and environmental communities. The retrieval of weather information is an important component of any planning where manned or unmanned flight is involved.

3.2.1.1 Environmental Effects on Weapons Employment

The effects of weather on air-borne weapons varies considerably with the weapon, but it is safe to say that there are no weapons, even iron bombs, that are totally unaffected by the weather. Non-powered precision-guided munitions (PGMs) such as JSOW are primarily affected by wind between the release point and the target. Powered-flight weapons such as JDAM are affected by temperature as well. Absolute humidity, precipitation, and aerosols affect weapons with IR seekers such as SLAM-ER.

Another weapon for which environmental effects are known to some extent is Tomahawk. In a previous study [10] by Daniel H. Wagner Associates, Inc. on the effects of near-real-time weather data into TLAM mission planning, we concluded that substantial benefits in range and TOT would accrue from the use of accurate weather data. Other studies [11] since then have quantified the TLAM requirements and the effects of various levels of weather phenomena on mission success.

Until recently, there has been little enthusiasm in the TLAM community for incorporating weather data into mission planning, because of the TLAM concept of operations, which involves planning missions days, weeks, or even months in advance of execution. In those circumstances, accurate predictions of weather conditions to be encountered at the time of the mission are not possible, and little benefit was seen to using, say, climatological data in place of standard day types. This circumstance has changed dramatically with the advent of the Afloat Planning System (APS) and Tactical Tomahawk Weapons Planning System (TTWPS) allows missions to be planned on-site in the theatre of operations. TIWPS Tactical Tomahawk missions to be planned directly at the aboard the shooter, for immediate execution. This

presents a recognized and immediate opportunity for the incorporation of weather prediction data into Tomahawk mission planning.

3.2.1.1   Availability and Accuracy of METOC Data for Mission Planning

Forecasts are generated by qualified Meteorological and Oceanographic (METOC) personnel on shipboard based on numerical forecast data originating from the Fleet Numerical Meteorology and Oceanography Center (FNMOC) and by regional METOC Centers. These data are "gridded", meaning they contain predicted weather parameters (e.g., wind direction and strength, temperature, altitude of 500 mb pressure level) at the vertices of a rectangular grid. Other direct and inferred weather products such as satellite imagery and weather conditions derived from satellite analysis are available through the Tactical Environmental Support System (TESS/NITES), which is located at many Navy afloat and ashore sites.

All these data, reviewed as appropriate by on-site METOC officers, as well as TESS/NITES products, are available to METOC personnel and other systems via the Tactical Environmental Data Server (TEDS). The information contained in the data is generally distributed to non-METOC personnel through manually-generated Horizontal Weather Depictions (HWD), text messages, and briefings.

At present, FNMOC generates 12-hour, 24-hour, 36-hour, etc., analyses and forecasts using the Navy Operational Global Atmospheric Prediction System (NOGAPS) [12] and Navy Operational Regional Atmospheric Prediction System (NORAPS) models. These models are "synoptic" in the sense that they capture the large-scale weather effects and ignore the small-scale ones. The synoptic nature of NOGAPS and NORAPS makes them of limited use for predictions of weather conditions at specific points, such as at the position of a PGM target. Also, for reasons internal to the models themselves, the effects of observation data that differ from the model values are purposely reduced [13]. This is a practice that preserves the overall model stability but clearly decreases its validity at the point in question, since after the update the model is maintaining a significantly different value than the actual observed one.

A critical problem with forecasts based on symmetric model data is that of necessity their times of validity ("tau") are spaced at regular intervals: 0000Z and 1200Z each day. Even if the forecast conditions were perfectly accurate for a specified grid point at, say, 1200Z, it is not clear what that forecast says about conditions at the same location at, say, 1600Z. While this problem is somewhat solved for operations offices by HWDs and weather briefings, the impossibility of accurate interpolation of gridded data prevents automated use of existing METOC products into mission plans.

A new mesoscale model, the Coupled Ocean/Atmospheric Mesoscale Prediction System (COAMPS), remedies the tactical shortcomings of the synoptic models, and should meet PGM tactical requirements for numerical forecasts [14-15]. COAMPS uses NOGAPS or NORAPS outputs to initialize and maintain its boundary values, provides high geographic resolution (1 - 7 Km between grid points), includes a detailed vertical profile (commonly with 15 sigma-levels), and takes into account important features of the lower atmosphere such as terrain effects and vertical dynamics. COAMPS is hosted in the Tactical Atmospheric Modeling System-Real-Time (TAMS-RT) which is being deployed to METOC Centers. TAMS-RT will allow easy integration of on-scene data from radiosondes, foreign and commercial sources, and pilot reports. It will also produce forecasts for any specified time, on demand. These capabilities, which have not previously been available, are critical to PGM mission planning and execution.

The NOGAPS and NORAPS models have been thoroughly validated using statistical comparisons of their predictions to the actual conditions. However, some care must be taken

in dealing with weather-prediction model accuracy, for several reasons. First is the variability of the data available to the numerical prediction models. Some regions, such as the subarctic regions of the Northern Hemisphere, have a very high density of data collection points, yielding correspondingly high prediction accuracy. Others, such as the South Atlantic, have much lower data collection densities, so the models are not updated and corrected as well. Another factor is the relationship of the statistical accuracy of the model at its specified grid points to accuracy at a single given point away from the grid points at a time other than tau. All the model evaluations we have seen compare the forecast conditions at their grid points at time tau to the actual weather at those points at that time (sometimes smoothed to reflect the model's synoptic nature). While these figures are indicative of overall accuracy, they do not speak to the question of how useful a forecast for a particular point is in predicting the weather at that point, or, more critically, at a point 20 miles away at, say, two hours after tau.

COAMPS inherently overcomes many of these problems. Its fine grid reduces the distances over which predictions must be extrapolated, and its terrain sensitivity helps account for variations in that respect. Because the model is run at a local site in support of the tactical units that use its data, forecasts valid at tactically-critical times can be generated (provided that there is a "pull" interface allowing the mission planner to specify the tau values he requires). On the other hand, COAMPS is relatively untested compared to the synoptic models. One of the activities being carried out by Naval Air Warfare Center-Weapons (NAWC-WPN) in support of our Navy Phase II development project is to collect data generated by COAMPS for recent operations and compare them to the actual weather at the forecast times.

Other sources of METOC data include the Air Force Global Weather Center (AFGWC) and Global Theater Weather Analysis and Prediction System (GTWAPS; not yet completed).

3.2.2   Objective 2: METPLAN Prototype Agent

In our Navy Phase II SBIR project, we developed two software products which are both related to the current research. These are:

- METPLAN, a software agent that automatically obtains available data on weather factors significant to a given PGM from the Tactical Environmental Data Server (TEDS), passes those data to the relevant mission planning system for incorporation into the mission planning process, and provides visual weapon-specific displays. Based on thresholds for high, medium, and low impacts set by the mission planner (or defaulted to weapon specifications), METPLAN displays the probable METOC effects in red, yellow, or green, respectively, to assist planners in building JSOW mission plans that are optimized for the predicted conditions.

- The Wagner Demo MPM, a mission planning module (MPM) mockup for the Joint Standoff Weapon (JSOW). The demo MPM accepts data from METPLAN, adjusts the planned route to account for predicted winds, and generally demonstrates the handling, display, and integration of tactical meteorological data into an MPM mission planning process. The basic function of the Wagner Demo is to provide a vehicle for demonstrating METPLAN functions in a realistic setting.

METPLAN is a generic agent, suitable for use by any mission planning systems such as JSOW, SLAM-ER, JDAM, and Tactical Tomahawk. METPLAN is currently driven by a JSOW requirements definition table that specifies the environmental variables of interest and the timeliness and model requirements that the METOC data must meet. This concept can

easily be extended to other PGMs, and in principle, METPLAN could also be extended to support higher-level planning, such as Strike or Interdiction Planning.

A key feature of METPLAN is its automatic nature. In the demo version (which is not usually connected to the Internet), the required METOC data is downloaded from TEDS to the mission planning system with no operator action required other than launching the TEDS client task, using a button in the METPLAN control window. For the operational version, we envision that the TEDS client will be actuated automatically upon receipt of notice that a new plan is being developed. While the plan is valid (i.e., from the time it is first formed until a specified time after its scheduled execution), METPLAN will continually interrogate the TEDS database for more current data. After execution of the plan, METPLAN will expunge the data from the appropriate mission planning directory so that there is no possibility of using stale data in mission plans.

3.2.2.1   METPLAN Features

Throughout the METPLAN project, we were guided by a Warrior Product Team (WPT) composed of end-users (pilots) who are intimately familiar with the requirements for PGM mission planning, especially JSOW. In their periodic reviews of our progress, the WPT made several precepts very clear. First, Navy pilots have become accustomed to the Portable Flight Planning Software (PFPS) in widespread use throughout the Navy and Air Force, and they suggested that METPLAN should run in that system. Additionally, the Joint Mission Planning System (JMPS) project announced that PFPS functionality would form the basis of JMPS. Consequently, our software is built into the PFPS environment. Second, the WPT members wanted environmental data integrated into the planning tool without them having to take any action. Third, they want, as far as possible, to deal with weather in terms of its impact on the mission rather than in purely meteorological terms. Finally, if there are weather problems, they want to be able to analyze and remedy those problems with as little effort as possible.

In addition to talking with our WPT, we reviewed a number of mission planning tools, including TAMPS (and the JSOW MPM), PFPS, and the Special Warfare Mission Planning System (SWAMPS), to obtain integration, operator-interface, and display ideas. The resulting METPLAN user interface and displays are described below.

The key display techniques are Stoplight Displays and Point-and-Click inquiries. Figure 8 shows a JSOW route generated by our demo JSOW MPM with weather-related stoplight displays from METPLAN.
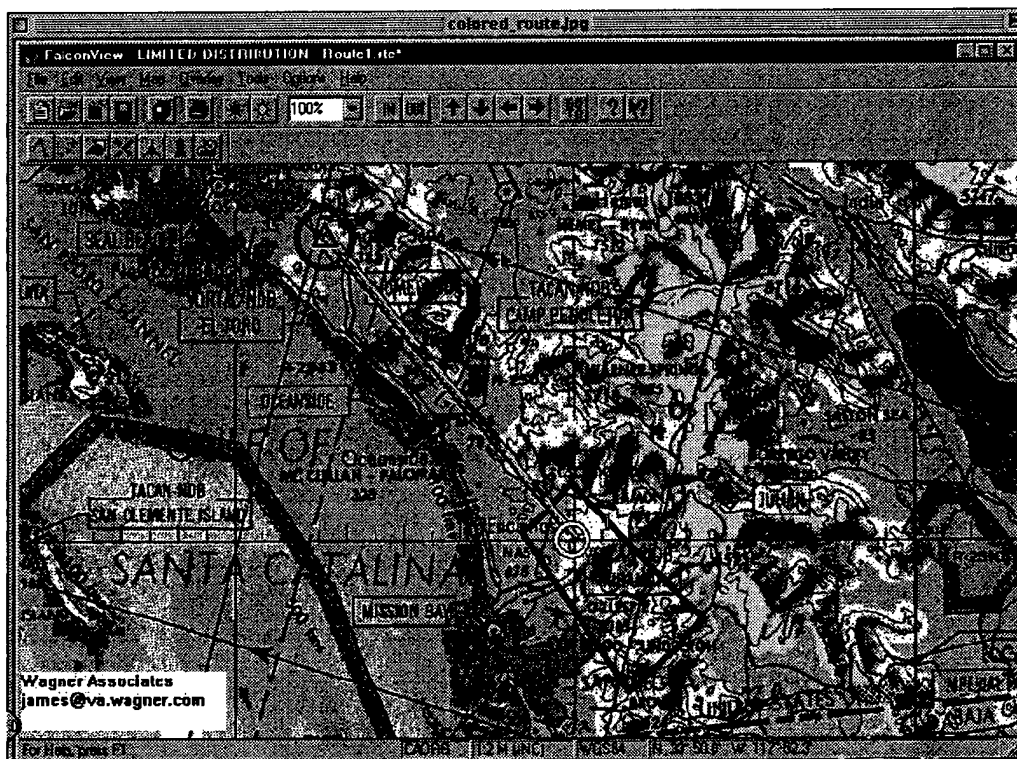
Figure 8. A JSOW Mission with Colored Route and Target Circle

Figure 8 shows a JSOW route for a strike against a target (triangular shape at top of figure) at El Toro, CA, from a weapon-release point (circular shape) southeast of the target. The symbols are generated by FalconView, the map server for PFPS, with which METPLAN is integrated. The large sector around the release point is the JSOW Launch Accessibility Region (LAR). The maximum range arc was calculated by the Wagner Demo MPM using the numerical weather forecast data extracted by METPLAN from TEDS on the day the plan was generated and valid at the planned time of missile launch.

The colored line displayed slightly to the right of the JSOW route indicates the weather conditions of interest to JSOW along that route. The red and yellow sections of the line indicate weather conditions that are above the corresponding thresholds in the Preferences menu. [1]

The colored circle around the target is a JSOW-specific METPLAN display. Because of the sensitivity of JSOW-A to wind strength and direction at the target, it is extremely useful for the mission planner to be alerted when the surface-level wind at the target is strong enough to cause a problem. The circular display in Figure 8 indicates that the wind is predicted to exceed the JSOW sensitivity thresholds, and is expected to blow from the southwest. The arc shows a 120-degree sector measured around the direct downwind approach direction. According to our conversations with JSOW mission planners, JSOW munitions performance degrades severely when the weapon approaches the target from that sector.

---

[1] Normally, red would be reserved for conditions severe enough to seriously impair the mission, but in this case the Southern California weather rarely exceeds such levels, and so the thresholds have been set artificially low in order to demonstrate the stoplight feature.

In the case of the target circle, a red sector can only mean that the wind speed at the target is predicted to exceed the specified strength and that the red sector indicates the unfavorable downwind approach angle. A red section of the route could indicate any one of a number of weather problems. To determine the problem, the user simply points to the section in question and clicks his mouse button.

Figure 9 shows the resulting display. The "Altitude" display indicates the planned missile altitude at that point in its flight: 25895 feet above mean sea level. The user has elected to look at "High Impact (Red)" factors, and observes that the wind exceeds the red threshold at all altitudes from 40000 feet MSL down to 25000 feet. Thus, he cannot expect to extend the range of his mission by changing the altitude of flight through this area. The wind direction of around 220 degrees gives him some insight into a possible longer-range prospect, however. If he can safely launch from the direction of San Clemente Island, he may be able to take advantage of the more favorable wind direction and gain a little stand-off range for his mission. He can check this by pointing to a representative spot between San Clemente and the target and clicking there. He will get the same display as in Figure 9, without the altitude figure. Note, however, that a direct shot from the San Clemente direction would take the JSOW through the red sector on the final approach, and inserting a waypoint to avoid this might more than offset the range gains attained by flying downwind during the early stages of the mission. Because these decisions involve tradeoffs such as flight range vs. probability of success, which in turn depend on tactical constraints and target properties, they cannot be made automatically for the mission planner.
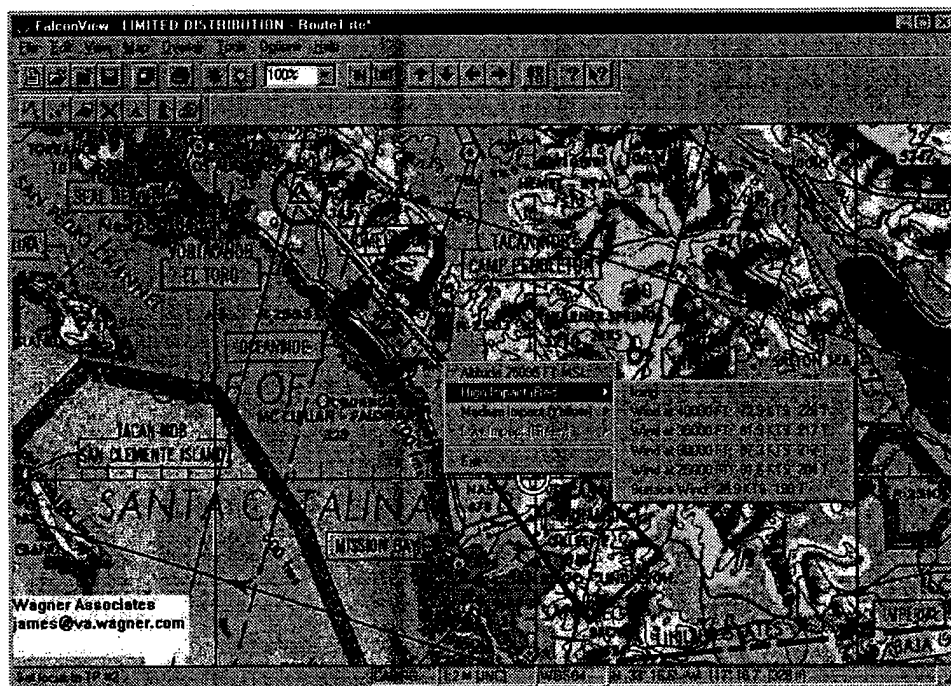

Figure 9.   Point-and-Click Weather Information

If the mission planner is interested in a slightly more detailed analysis of, say, the wind direction, METPLAN can supply him with graphical products from TEDS. Figure 10 shows an example product, wind streamlines. Our WPT indicated that mission-planning personnel find colored wind streamlines much easier to analyze quickly than the classical wind barbs. Note the effects of land features on the wind, as shown by the bending of the streamlines at the coast and the channeling along the coastal range just inland of Los Angeles. Wind streamline

products are not yet available from TEDS, so the underlying data for Figure 10 was downloaded from the Naval Research Lab-Monterey (NRL-MRY) website to illustrate the display.
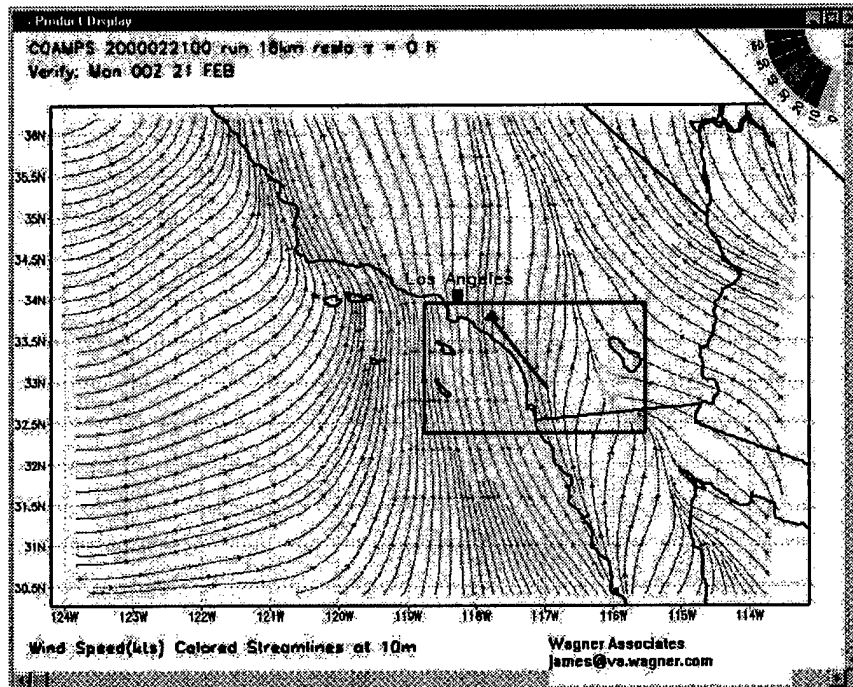

Figure 10.   Wind Streamline Display

Figure 10 shows the area of interest, as indicated by the geographical size and position of the tactical display in Figures 1 and 2, and the line and triangle in the center of that rectangle depict the JSOW route. This is an idea we obtained from SWAMPS. Giving the viewer a new window that displays the area of the tactical window allows him to see the features of interest without cluttering up the tactical display. It also allows a broader field of view than the tactical one, without the necessity of scaling a picture out and then back. Our WPT members indicated that they find such scaling disorienting as compared to viewing the weather product in another window.

The streamlines of Figure 10 were colored for wind strength by NRL-MRY according to an arbitrary color scheme shown in the upper right corner. This scheme, while pleasing to the METOC officer, bears no import to the Mission Planner. In the proposed operational METPLAN, we will build the streamlines from TEDS data using the same algorithms that generated the lines in Figure 10, and color them according to the thresholds for the particular weapon. This will provide streamlines (or other products) that have a direct bearing on weapon performance.

In the METPLAN concept of operations, the thresholds for red, yellow, and green coloring by METPLAN are changed rarely, if at all. They are set according to weapon sensitivity to each weather factor, and would only be modified in the event that the mission planners become aware of new information about those sensitivities. Figure 11 shows a set of wind thresholds selected for JSOW-A. in this case, the user has indicated that he wants to see the downwind-approach sector colored red if the predicted wind at the target exceeds 20 knots, and yellow if it exceeds 10 knots. He is interested in coloring the arc 60-degrees on either side of the direct downwind approach. He wants to see icing listed as a problem if it is predicted at

18

the JSOW's flight altitude, but is not interested in having precipitation or turbulence flagged. The choices on the right indicate different ways the colored route can be displayed; the choices shown in Figure 11 result in the effects shown in Figure 8.
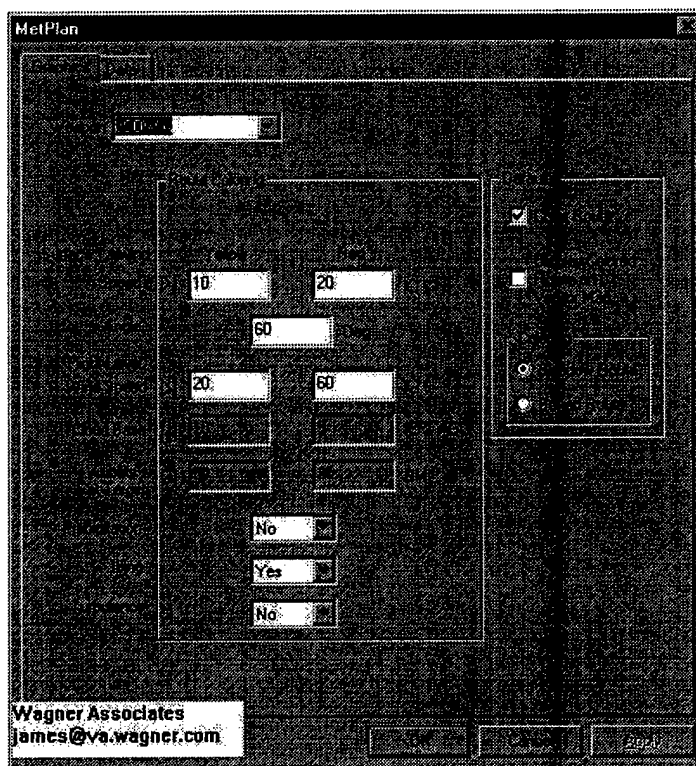

Figure 11. METPLAN Preferences Screen

Once a plan is initiated (actually, once a target has been chosen for a mission at a given time for a weapon), METPLAN searches the TEDS database for data appropriate to that weapon, valid at the appropriate time. The particular weather variables critical to that particular weapon have been defaulted as well, but the user is free to select other factors as he chooses. These factors will be added to the METPLAN Preferences screen as they are selected, and the user will be prompted to select appropriate thresholds for each. The METPLAN screen for selecting such additional factors is shown in Figure 12.

Our concept of operations for METPLAN includes a "time line" for each weather factor of interest. The METPLAN field selection screen shown in Figure 12 also allows the user to dictate the time window of interest. Here, the user has decided he wants to see all forecasts for the time interval beginning 1 hour before the scheduled mission execution time and extending 12 hours further into the future. *The button for "launch client" is only for use in the current version of METPLAN because the system is rarely online to a TEDS. This button will be removed in the operational version of METPLAN.*
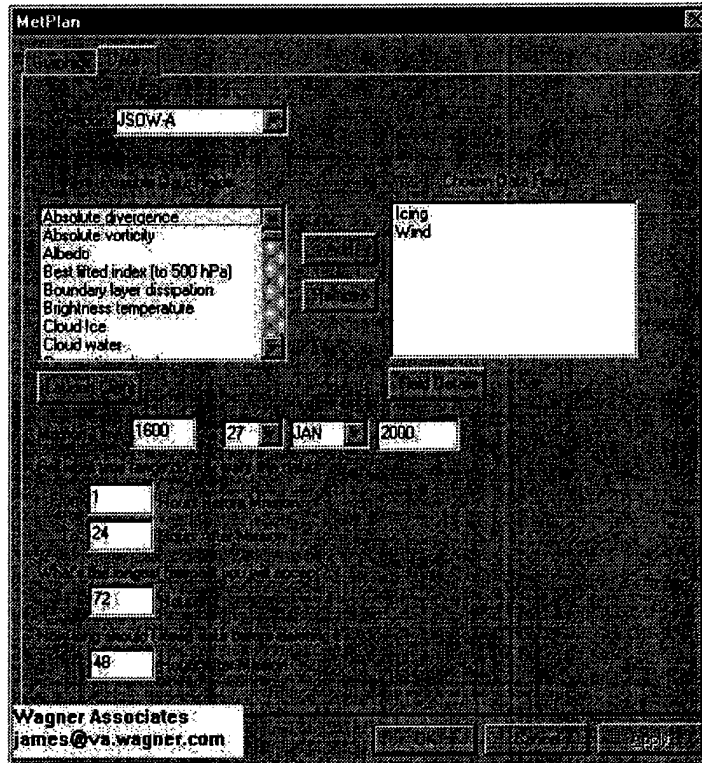
19

Figure 12.   METOC Field Selection and Control

The architecture of METPLAN is that it operates as a separate process along with the Wagner JSOW Demo, with both processes accessing FalconView to draw on the map. This architecture, which is expected to be utilized in JMPS, requires that each process be controlled through a separate user interface. The METPLAN control window is shown in Figure 13. The control buttons come in two sets. The set on the left deals with METPLAN products and integrated mission planning tools; the selected button shows the same route-planning logo as appears on PFPS/Wagner Demo, and indicates that the user wants the route colored according to the weather thresholds, as in Figure 8. If that button is deselected, the route is simply drawn as a black line.
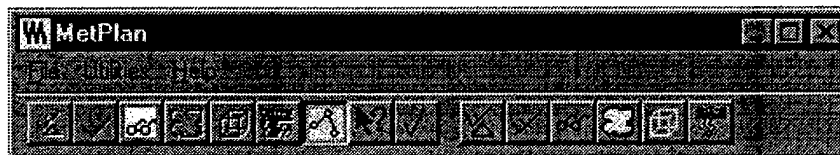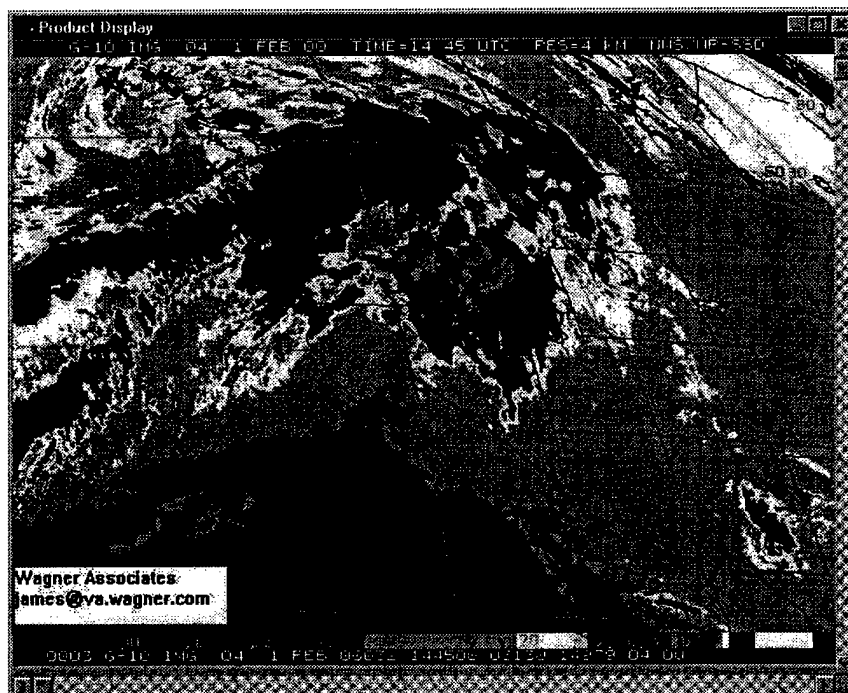

Figure 13.   METPLAN Program Control

20

Figure 14. Enhanced Satellite Imagery

3.2.2.2 METPLAN Architecture and Status

Conceptually, METPLAN consists of three parts: a TEDS client, which queries TEDS for the appropriate data; a MPM Server, which records preferences from the Mission Planning Module, puts the required data in the specified directories, and cleans up the data after the mission is concluded; a Graphics Server, which monitors the activity of the PFPS Route Server and displays the appropriate graphics on the tactical display as required; and the Graphical User Interface (GUI), which displays the METPLAN Program Control toolbar, the Preferences screen, and the Field Selection and Control screens (Figures 11-13), and records the user's controls and preferences for use by the other parts of the program. The version of METPLAN developed in Phase II of the Navy project lives entirely on the application side of the TEDS/Mission Planning interface.

The TEDS Client makes use of the published TEDS APIs to obtain data from TEDS. The TEDS APIs implement a client for the Informix database of TEDS, and provide routines to manipulate the data (for example, to extract a subset based on latitude and longitude constraints) once it is received from TEDS. The basic process carried out by the TEDS Client is as follows: (1) Get TEDS Catalog for weather variables of interest to the weapon system involved (JSOW); (2) Determine whether appropriate data is available for the time of the mission, and if so, which data is preferred; (3) Obtain the data from TEDS: (4) Subset the data for the region of interest; and (5) Manipulate the data as required for the weapon in question (e.g., calculate probability of icing from temperature and humidity).

Functionally, the Phase II TEDS client is quite complete, and would, with minor modifications, serve as an operational program to obtain data for any weapon system. However, PMW 185, the cognizant agency for METOC software, is opposed to having applications such as METPLAN use the TEDS APIs directly, since these APIs are being phased out in favor of METCAST queries. METCAST, a TEDS data-access program completed and fielded during the course of this project, uses HTTP for remote access, rather than allowing direct INFORMIX client calls into the database. In the view of PMW 185,

21

METCAST can provide a single, generic interface to TEDS, a solution that improves control over TEDS access and reduces the expense of changes to the TEDS data structures. The METCAST approach has the added benefit that it uses the HTTP port, which is almost always open in a firewall-protected system (otherwise there would be no internet access for the users of that system), instead of the INFORMIX port, which is generally not open. In the remainder of Phase II of the Navy project, we are modifying the TEDS Client of METPLAN to use the METCAST structures instead of the TEDS APIs.

Of the four components of METPLAN, the least-developed is the MPM-specific Server, for the simple reason that in Phase II we only served mission planning for one MPM, JSOW, and for one plan at a time. In that context, the server need only refer to the user preferences and controls, which, along with the time and place of the mission define the weather data required. The directory in which to place the data for MPM use is always the same.

The METPLAN Graphics Server is complete, for the purpose of displaying stoplight displays for JSOW, point-and-click displays for any weapon, and overlays and other products (e.g., wind streamlines) for any application. METPLAN's overlaying technique is to shade polygons with the appropriate color for the particular variable and the thresholds for that variable.

3.2.3   Objective 3: Prototype Agent Wizard

METPLAN is an example of an agent that performs certain tasks, including data retrieval, display, and management, for specific users and clients. The principal objective of the current research is to design an Agent Wizard that a user can employ to create new agents with different functions.

Phase I prototype is a skeleton of just such an Agent Wizard, with tools and processes oriented toward environmental data. The prototype is developed in C++, although the Phase II product we envision will be in Java. The design, as represented by the prototype, has the following features.
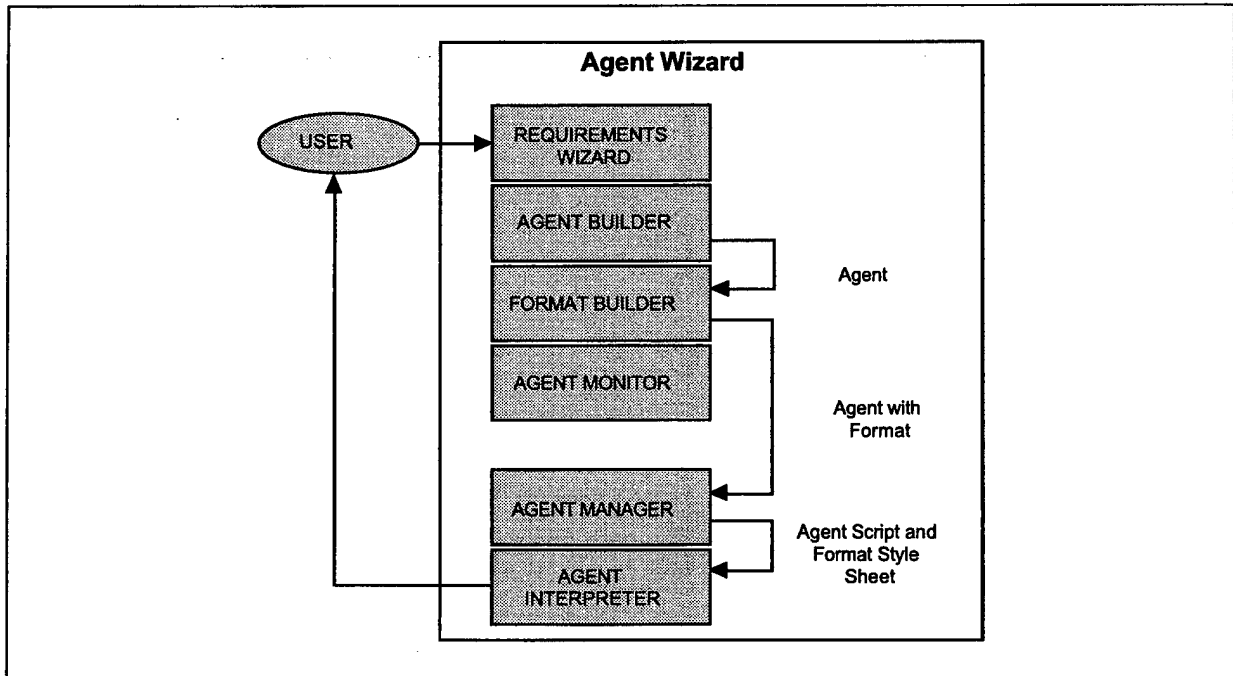
Figure 15.   Agent Wizard Architecture

3.2.3.1   Requirements Wizard

The requirements wizard interacts with the user through a Graphical User Interface and permits the user to specify types, sources, and uses of the data to be retrieved by the agent.

3.2.3.2   Agent Builder

The Agent Builder creates agent scripts and objects, including the following types of agents:

Agent Wizard Language Scripts. The Agent Wizard is constructed using its own internal language. When a user creates an agent to be managed through the GUI, the Wizard generates a script in that language and then interprets and executes that script directly.

External Language Scripts. As an option, a more advanced user, such as an application developer, may wish to create transportable agent objects and for reasons of architecture and expertise, may require that these objects be written in a specific language. Languages we intend to support in the Phase II prototype include Perl Scripts, Java classes, JavaBeans, and C++ classes.

3.2.3.3   Format Builder

The format builder, through the GUI, helps the user to specify the format of the data to be presented after retrieval. The format builder will have knowledge of formats commonly used for the domain. The underlying technology of the Format Builder will be the eXtensible Stylesheet Language (XSL), which provides formatting capabilities for XML.

23

3.2.3.4 Agent Monitor

The agent monitor communicates with each agent and collects and displays progress data through the GUI. This progress display is more than a simple thermometer, and displays the steps in the agent's task list, sources of data, and progress toward retrieval or display.

3.2.3.5 Agent Manager

The agent manager provides coordination among multiple agents, to prevent duplicate retrieval of data from external data communications channels. This way the system can avoid overloading of network links and at the same time provide quicker response to users, in cases where multiple agents are retrieving the same data.

Whenever an agent needs certain data, it first checks with the agent manager. If that data is already present on the local network, the agent manager provides the storage location to the requesting agent. The agent manager can also prevent an agent from deleting data after use, if that data is still in use by another agent.

The agent manager also manages external agents, i.e., those created in external languages by the local Wizard or Wizards at other locations. Every agent created by the Wizard will automatically "ping" the local network to see if there is an agent manager running. If there is, the agent will subscribe to the manager and publish its data, as though it were an agent created locally.

3.2.3.6 Agent Interpreter

The Agent Interpreter executes the Agent Language Script that defines an agent's behavior (e.g., "retrieve gridded data (wind, temperature) from TEDS-NRL for 1200Z July 12, 2000 in San Diego").

The agent interpreter applies the data format style sheets to the retrieved data and displays the data appropriately to the user. The interpreter can independently display data, invoke a browser to display text or graphics, overlay data on map servers, and send data in specific formats to other running processes.

3.2.3.7 Graphical User Interface

For the Phase I prototype, we built a Graphical User Interface (GUI) that demonstrates how the users will create agents using the various components of the Wizard. The figures that follow show the various stages of wizard creation and management using this GUI.

Figure 16. User Profile and Logon Screen

Note: The user enters a name, selects a knowledge domain and knowledge level. If there is a profile, the user may use it or he may create a new one. Each user's profile determines the look and feel of the GUI as it is displayed to him. For example, an application developer sees the "Translate Agent" button on the Agent Builder (see Figure 23), but a general user does not.
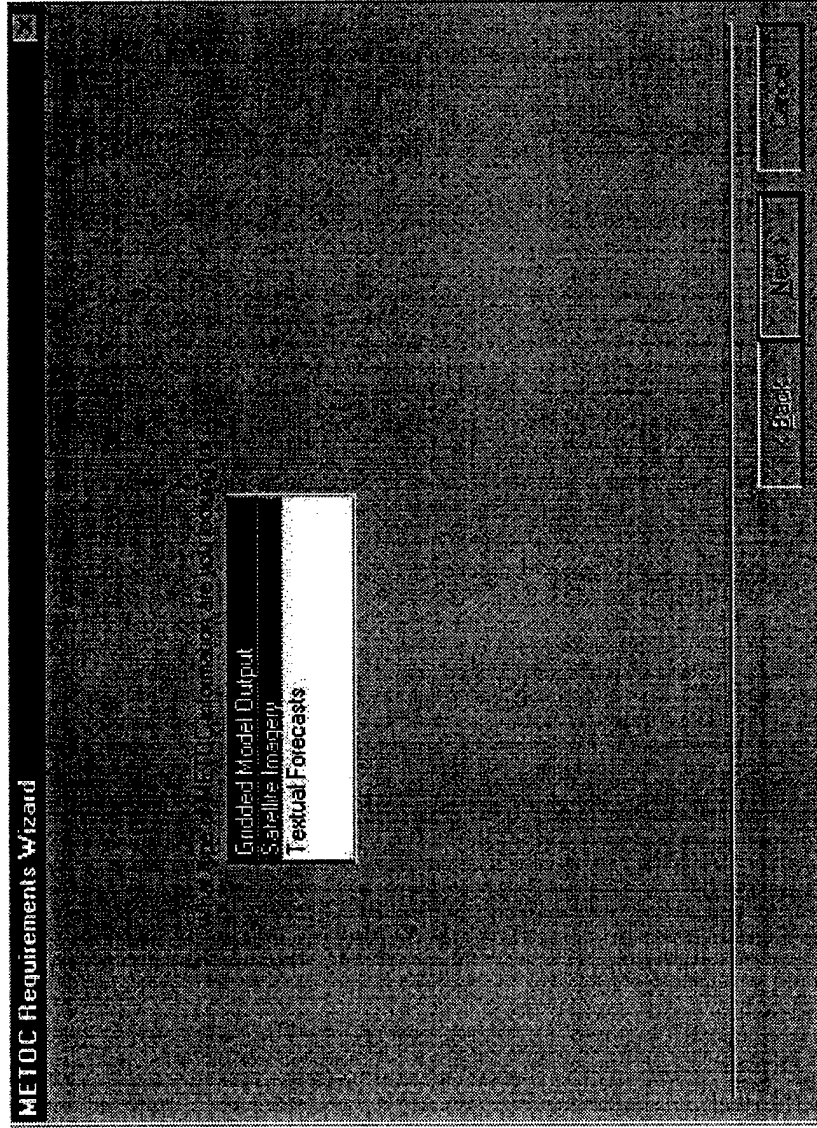
25

Figure 17. METOC Requirements Wizard

Note: The Phase I prototype Requirements Wizard is oriented toward METOC information requirements. The wizard screens shown in Figures 17-22 step the user through a subset of the information needed to meet general METOC requirements.
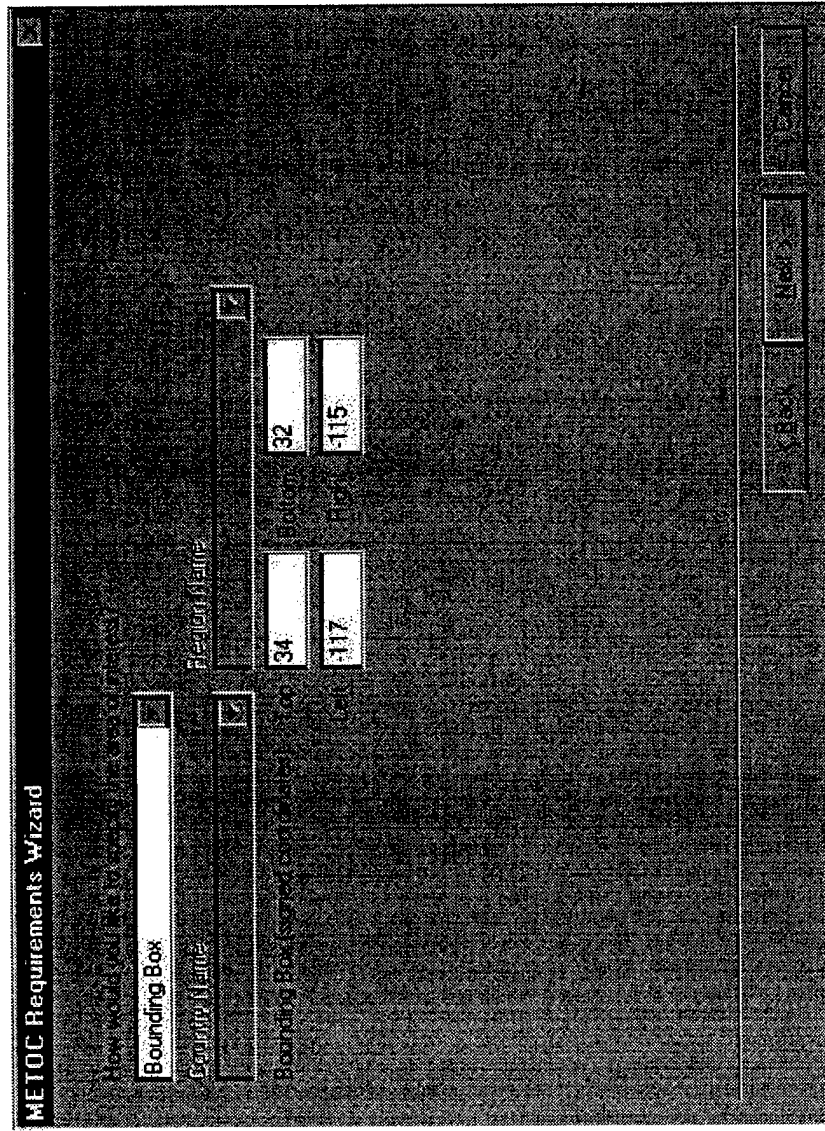
26

Figure 18. Defining the Area

Note: Building the Requirements Wizard toward a specific data model (e.g., METOC) allows us to provide the end-user with more user-friendly features. This screen shows how we can improve on the general Bounding Box by providing mappings of lat/lon boxes to country and region names.
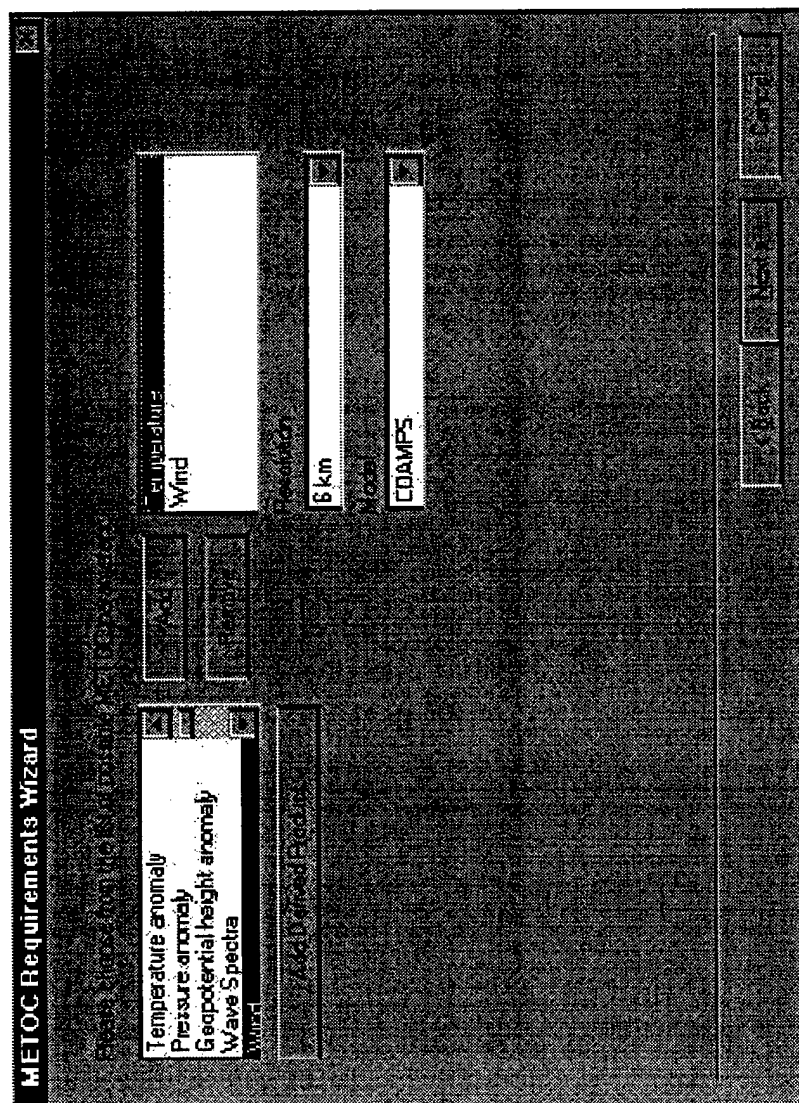
27

Figure 19. Choosing the METOC Parameters

Note: On the left hand side of the screen is the list of METOC parameters taken from the World Meteorological Organization's (WMO) Grid in Binary (GRIB) standard. The user has the option of defining his own derived product from these core parameters (e.g., defining the formula for modified refractivity as a function of temperature, pressure, relative humidity, and height). Since we've chosen to retrieve gridded binary data (Figure 17), we see here the options for specifying which model and resolution fit our needs.

28

Figure 20. Times for Data Retrieval

Note: This screen captures the temporal extents of the required data and the agent's behavior. If the user chooses to have the retrieval information updated over time, the Agent Manager will re-run the agent at the requested times for as long as the user desires. The user can then control the agent (pause, stop, re-direct, etc.) through the Agent Monitor (Figure 28).
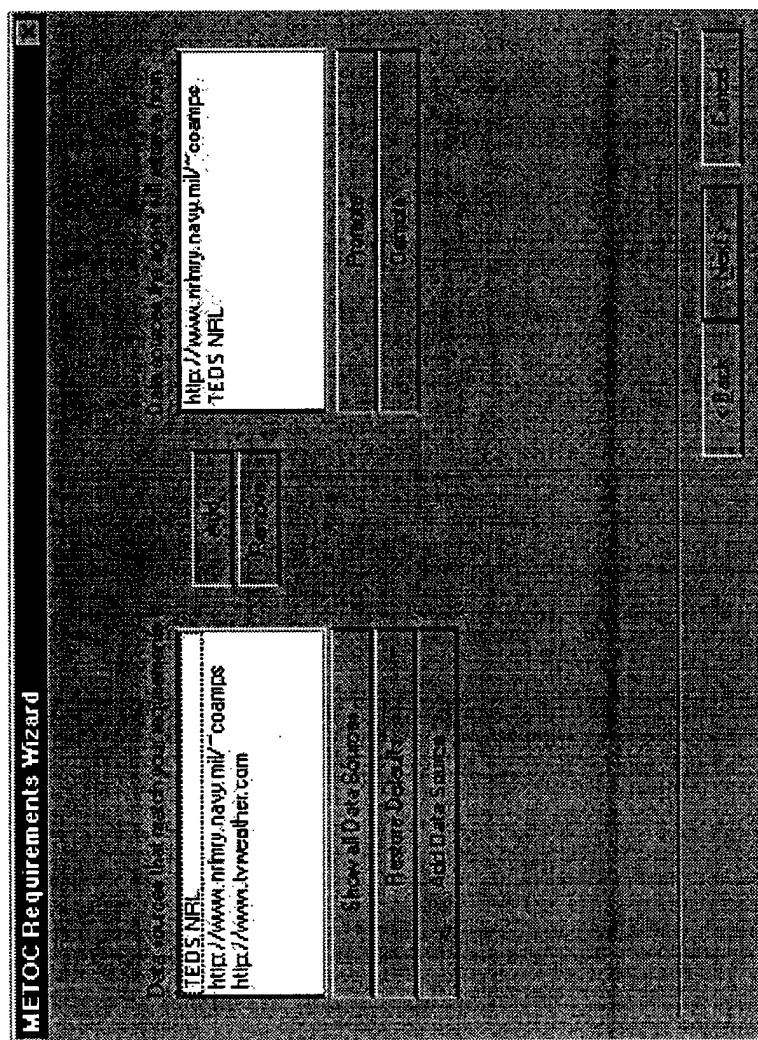
29

Figure 21. Choosing the Data Sources

Note: Underlying the Agent Wizard will be a distributed object management system, which will provide dynamic resource discovery. In the military, the Data Exchange Infrastructure (DEI) will provide the METOC domain with object management. In other military domains and the commercial sector, there will be some CORBA-like architecture for naming the data sources on the network. At the very least, data sources can be identified by their Uniform Resource Indicators (URIs). This screen allows the user to choose from the available data sources that match the requirements given in the wizard. The user has the option of ignoring the wizard's assessment of his requirements by clicking on "Show all Data Sources." He can then revert to the wizard's recommendations by clicking "Restore Defaults." Promotion and demotion of data sources affects the order in which the agent performs its retrieval task.

30
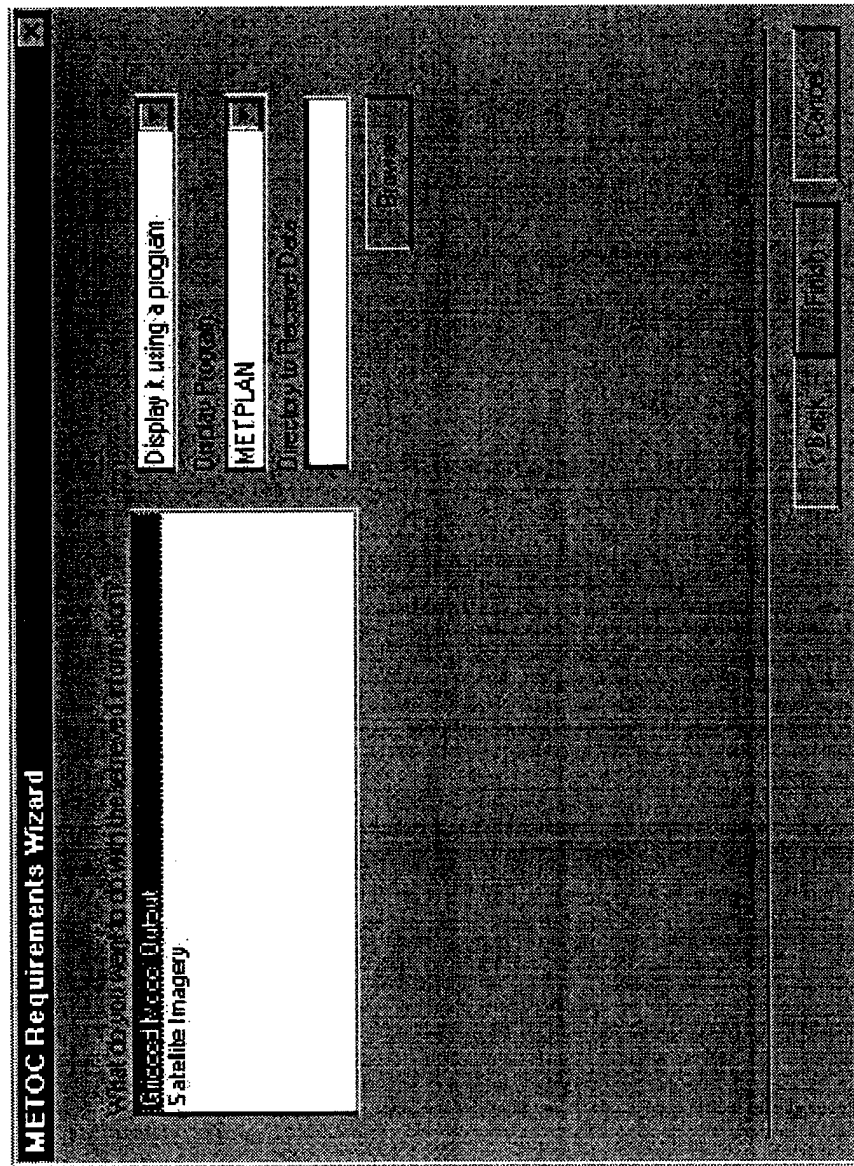
Figure 22. Receiving Process for Retrieved Data

Note: The user has the choice of storing the retrieved data in a local or networked directory or displaying the data using a program. In the example above, we've chosen to display gridded model output in METPLAN. This will feed METPLAN with the data it needs to provide the stoplight displays shown in Figure 8. Another option would be to display satellite imagery in Netscape or Internet Explorer.
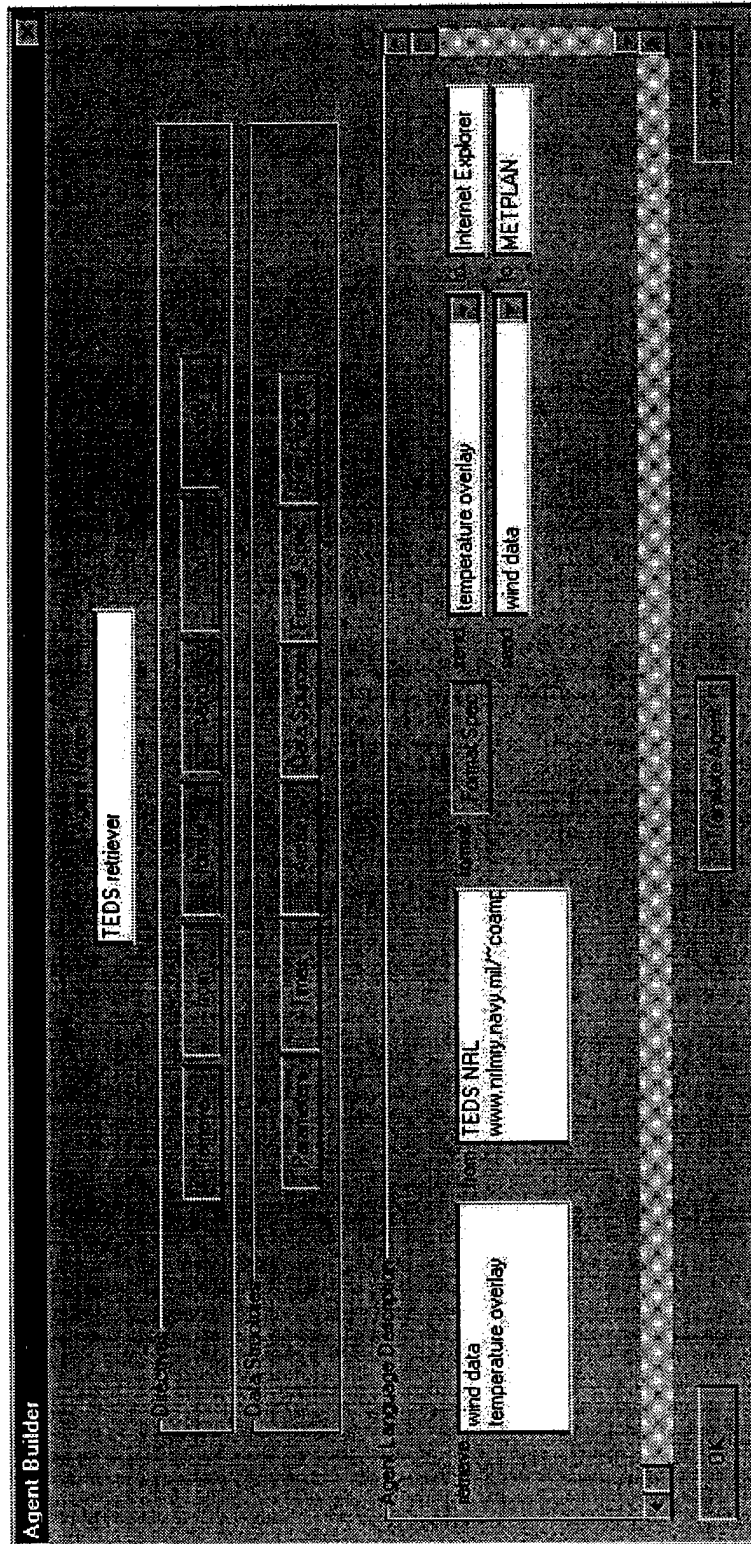
31

Figure 23. DEI Agent Builder Screen

Note: Once the requirements have been defined, the Agent Wizard creates an Agent Language Script definition of the required agent. The advanced user has the option of modifying any piece of the agent definition, while the novice user can simply click "OK" and let the agent do its job. The Directives section shows the Agent Language (currently a subset of what it will be in Phase II). The Data Structures section provides access to the same screens as in the Requirements Wizard to allow the user to change his requirements definition. This version of the Agent Builder assumes a heavily METOC-oriented infrastructure on the network (such as DEI). Phase II work would include the total configuration of all screens based on user profile (e.g., a general user could choose never to see this screen).

32
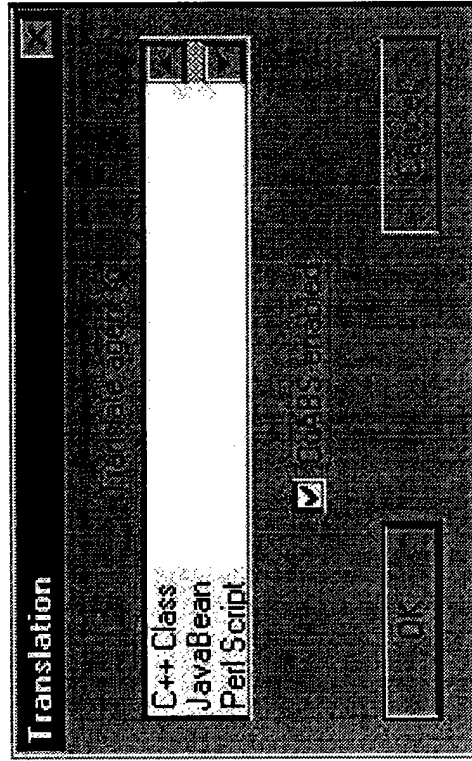
Figure 24. Language Translation Screen

Note: Clicking on "Translate" in the Agent Builder allows an advanced user (e.g., application developer) to export the agent definition as a C++ class, JavaBean, or Perl Script. The user can then run the agent stand-alone or integrate it into other software that requires the agent's particular expertise. In Phase II, selecting the "CoABS-enabled" check-box will cause the Agent Builder to provide the agent with the Java wrapper necessary to communicate with the CoABS grid. This will allow the agent to communicate with and be controlled by other agents and end-users on the grid.

33

Figure 25. CORBA Agent Builder Screen

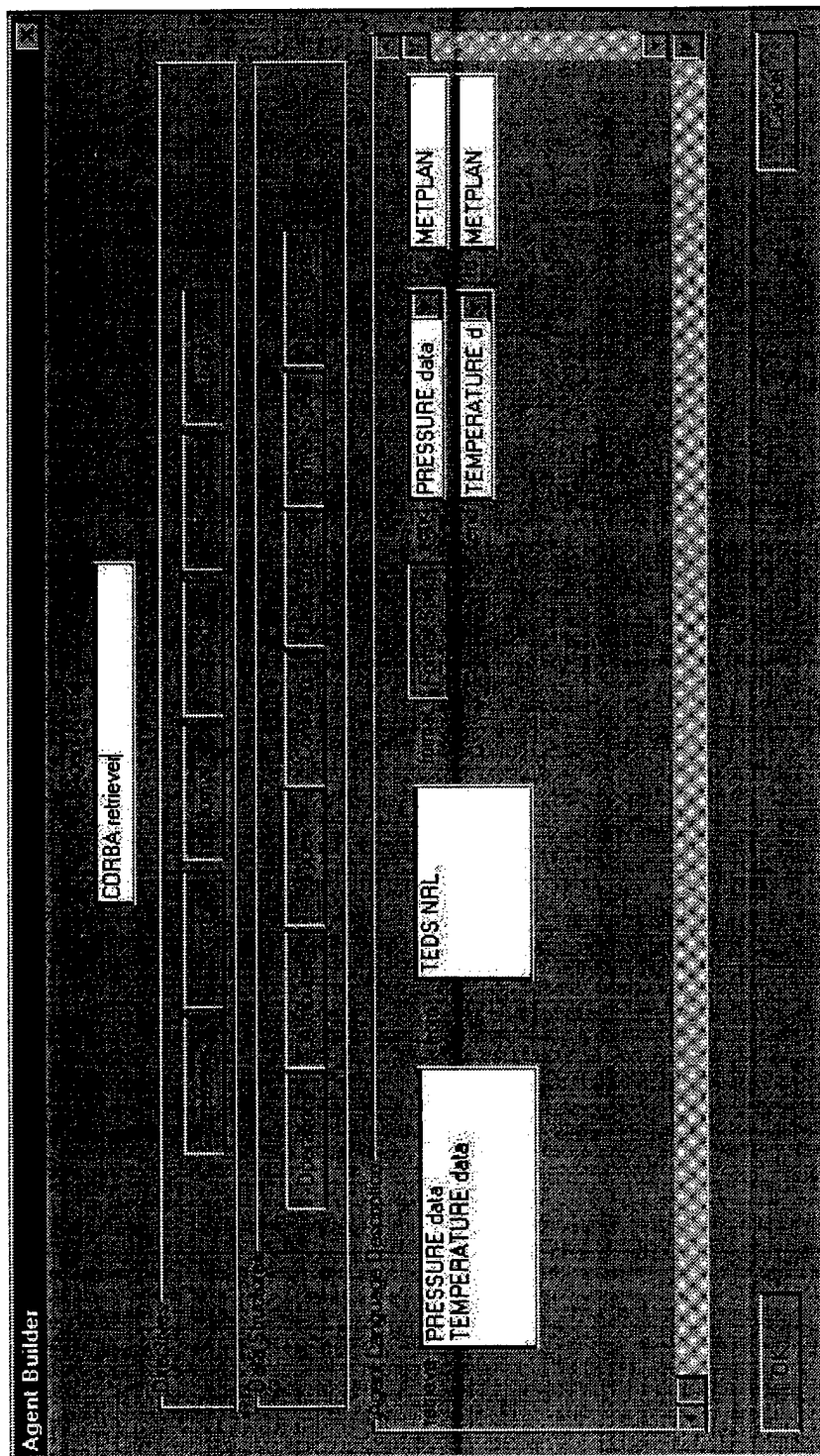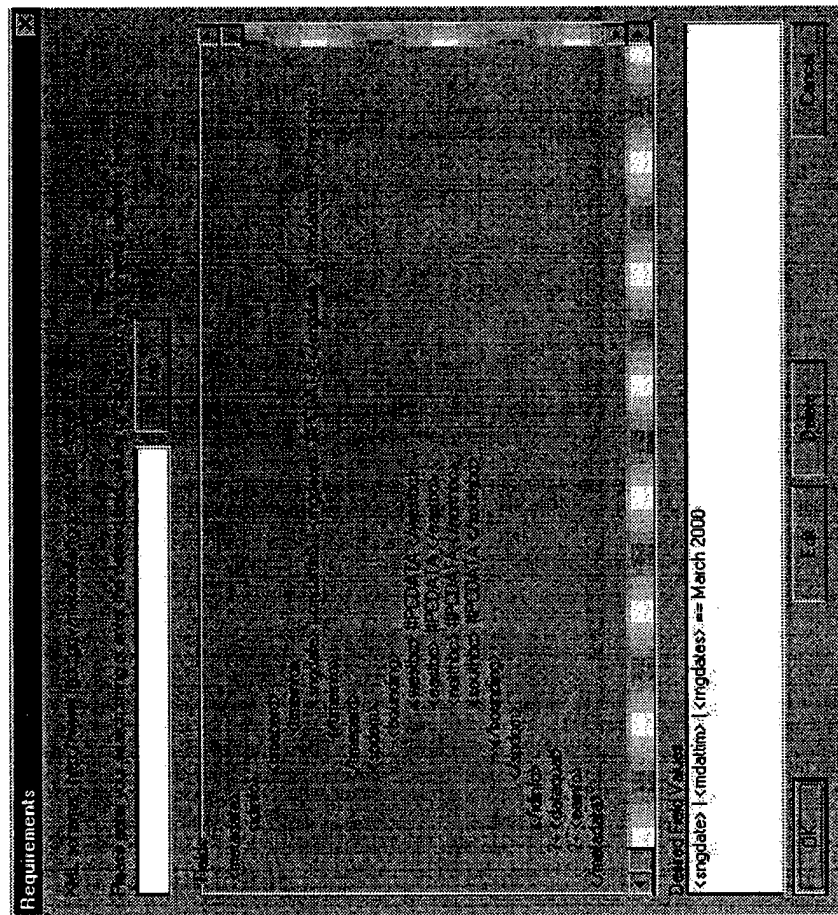Note: If a heavily METOC-oriented infrastructure such as DEI does not exist on the network, but a CORBA infrastructure does, the Agent Builder will look like this figure. The standard CORBA Operations, Attributes, Types and Constants have replaced the domain-specific data structures shown in Figure 23.

34

Figure 26. XML Requirements Builder

Note: The Phase I Prototype Requirements Wizard is oriented toward the METOC knowledge domain, but we envision the emergence of agent requirements in many other knowledge domains. In Phase II, we plan to build domain-specific Agent Wizards for METOC and other domains. However, we also want to provide the means by which the end-user can create agents for currently unknown knowledge domains. To that end, we have designed the XML Requirements Builder. Given an XML schema, the Requirements Builder allows the user to provide search criteria for the elements of the schema. The agent created to meet this requirement queries its data sources for matching schema element values.

35
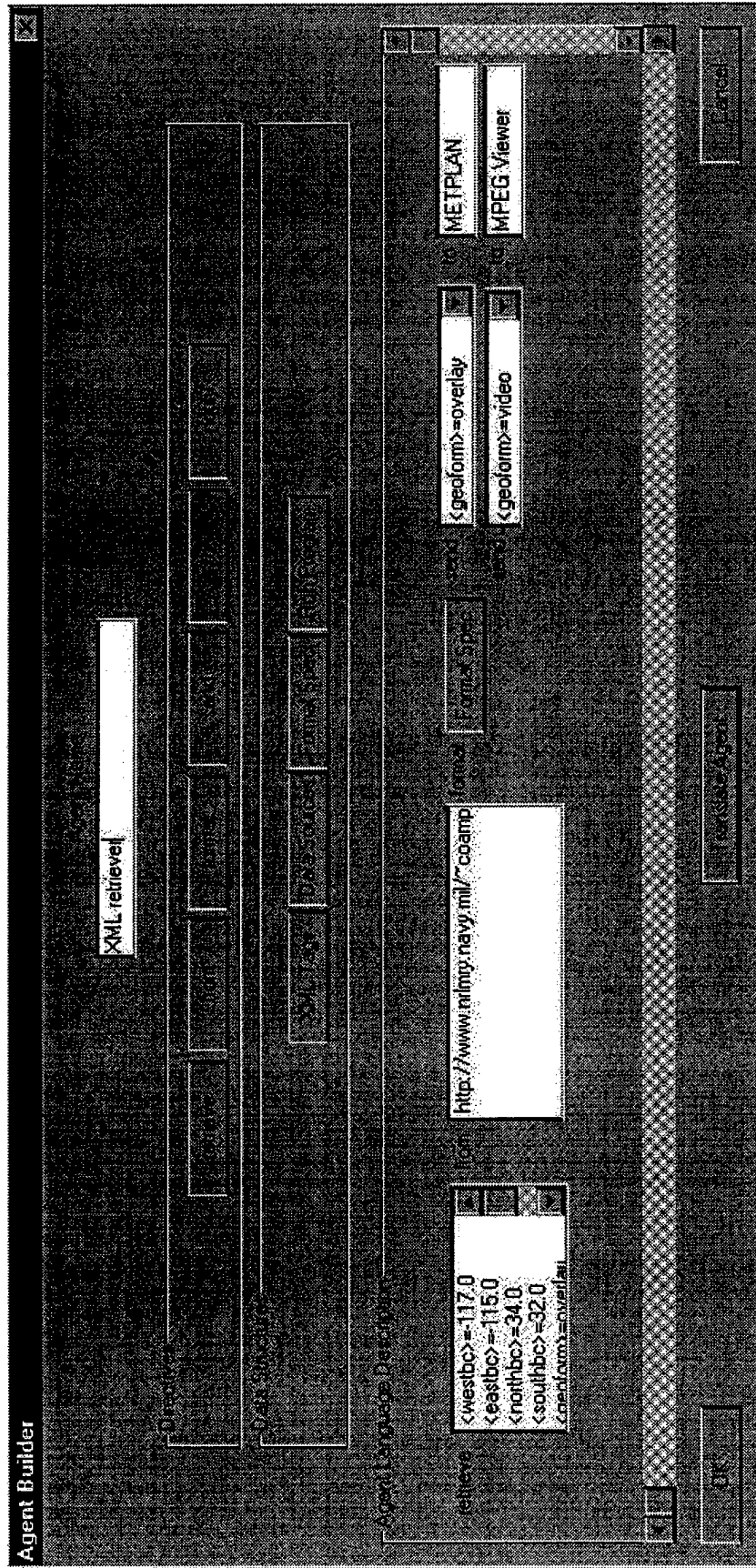
Figure 27. XML Agent Builder

Note: This screen is nearly identical to the DEI and CORBA Agent Builders, except that the Data Structures have been replaced with XML metadata tags.

36
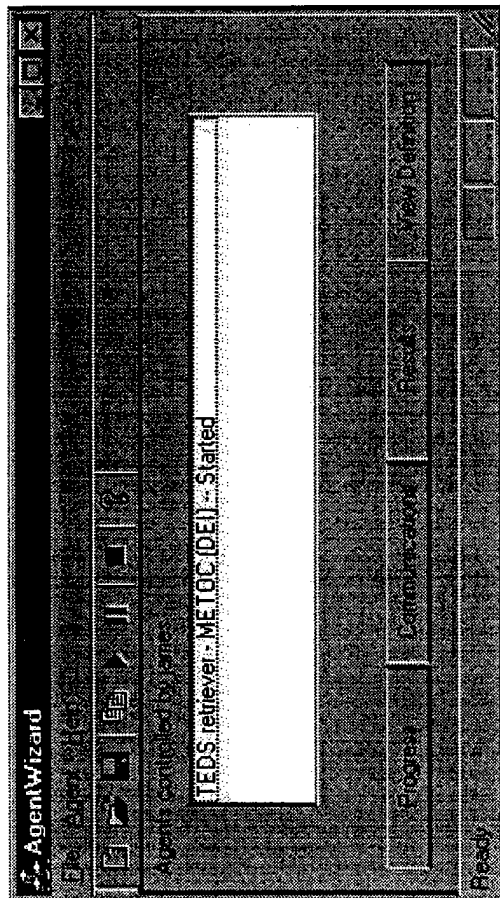
Figure 28. Agent Monitor

Note: Provides view and control of all agents created by the current user. Phase II enhancement: allow the user to view the behavior and definition of any agent on the network. The ⬚ button allows the user to clone a selected agent and modify it to meet new requirements. ▷ starts the selected agent; ⬚ pauses the selected agent; ⬚ stops the selected agent.

37
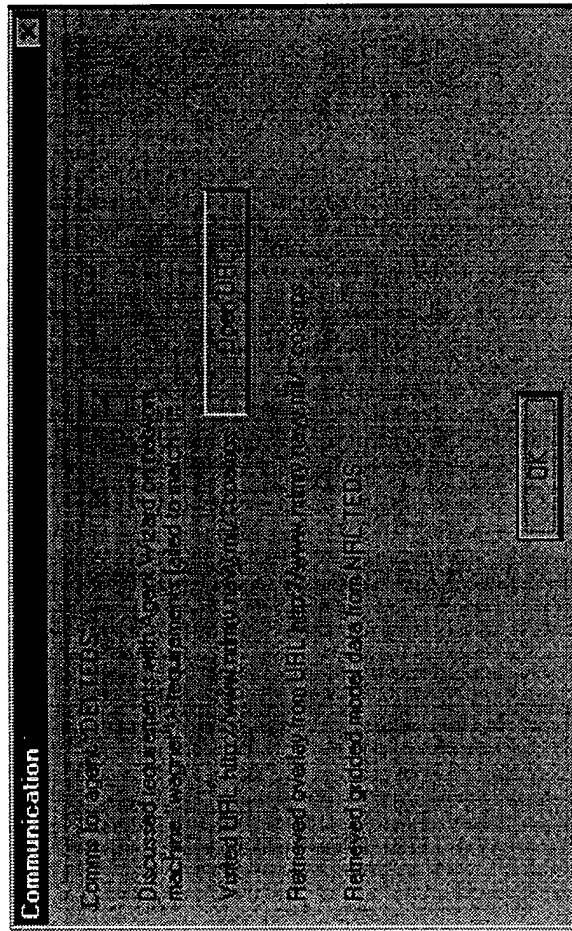
Figure 29. Reporting Agent Manager Communications

Note: This screen provides the user with a view of all communications between this Agent Manager and other Agent Managers on the network on behalf of this agent, as well as all communications between this agent and external data sources. When the agent visits a web site, the user has the option of opening that site to determine its validity or current state (e.g., server down, hasn't been updated in two years, etc.).

38
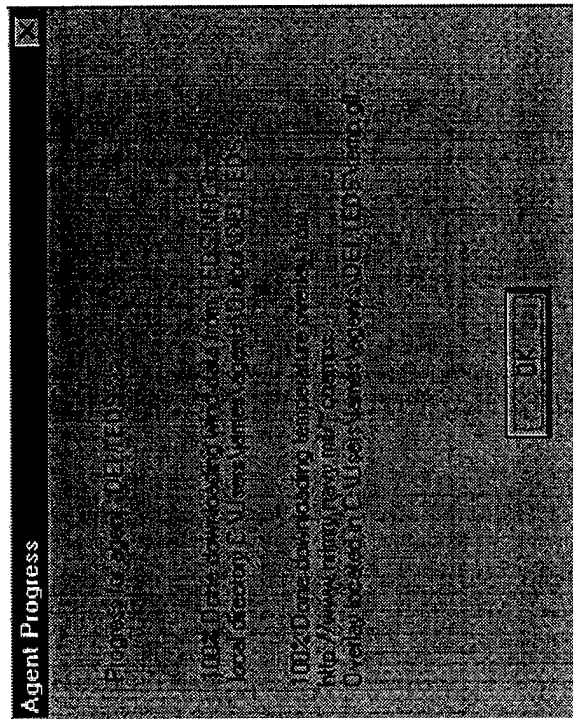
Figure 30. Agent Progress

Note: This screen provides a snapshot of what the agent has accomplished.

39
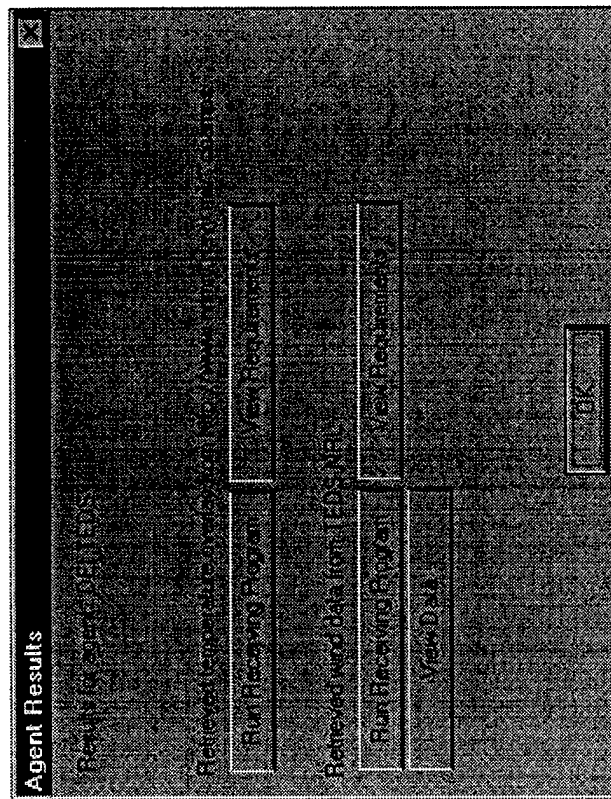
Figure 31. Reporting Retrieval Results

Note: This screen provides the user with the means to view and/or use the retrieval information (e.g., run METPLAN with the new data). Each result has a "View Requirements" button to link the results with the requirements that led to the results.

40

3.2.4 Objective 4: Object Schemas and Architecture

In the prototype we showed how the Agent Wizard could create agents using data models from many different sources. These data models must be understood by the Wizard in advance, in order to permit the construction of the required agents. It will also be possible to use the Wizard when *no* data model is available. In such cases, the wizard will be able to retrieve XML definitions from web pages and other sources and present those definitions to the user. Under this approach, the user would be able to select elements from the XML definition and use those elements in a script, thus creating a "free-form" agent to retrieve and manipulate data from that source.

3.2.4.1 Data Exchange Infrastructure (DEI)

For the DOD customers in Phase II, our most promising architecture appears to be the Data Exchange Infrastructure (DEI), being developed within the Navy to standardize access to all environmental databases. Figure 32 shows how the Agent Wizard will interact with the DEI world.
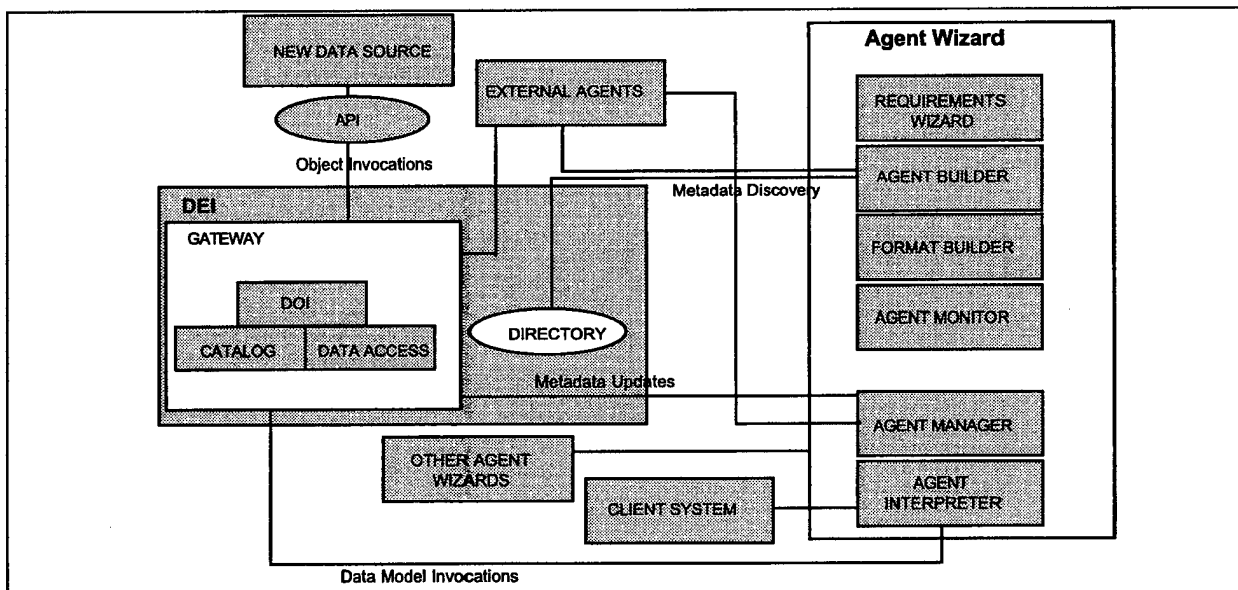

Figure 32. The Agent Wizard in the DEI World

DEI is an enhanced Common Object Request Broker Architecture (CORBA) infrastructure in which each data source is accessed through a Gateway. The Gateway is an enhanced Object Request Broker (ORB), containing three distinct interfaces: the Catalog, which lists the information available from the data source; the Data Access interface, which provides pointers to the underlying data and its full Data Model representation (metadata); and the Data Object Interface (DOI), which provides translations into common formats. Builders of new data sources will make use of the Gateway Development Kit, which will be one of the deliverables of the DEI project. DEI will provide a directory service for describing the gateways available on the network.

Based on our continuing discourse with John Ellis, Suan Starke, and Richard and Robert Owens of NRL, it appears that the DEI and Agent Wizard projects complement each other very well, with very little overlap. For example, the Agent Wizard will enable users to build intelligent agents easily that can be used to obtain data through DEI. Providing users with the ability to build these agents easily should result in significantly more use of DEI and of the

41

METOC data that DEI can be used to provide. In addition, the Agent Wizard can provide user services required by DEI, such as dynamic formatting/conversion of retrieved data for use by client programs. On the other hand, DEI will provide an enhanced CORBA environment that will allow the Agent Wizard to operate with minimal user input, and, in particular, without any inputs concerning the structure of the databases that are being accessed.

The Data Model used by DEI is still a work in progress, but it will be based on the National Oceanographic and Atmospheric Association's (NOAA) and Environmental Protection Agency's (EPA) current data model work, IBM Data Explorer's data model, the Synthetic Environmental Data Representation and Integration Specification (SEDRIS), and the Content Standard for Digital Geospatial Metadata (CSDGM) from the Federal Geographic Data Committee (FGDC). Our assurance from DEI is that the Data Model will fully describe data sources so that agents will be able to glean understanding from their metadata to meet user requirements with minimal user interaction required.

3.2.4.2 Common Object Request Broker Architecture (CORBA)

For the CORBA sources, a similar agent wizard interaction is possible, using a slightly different approach, as shown in Figure 33.
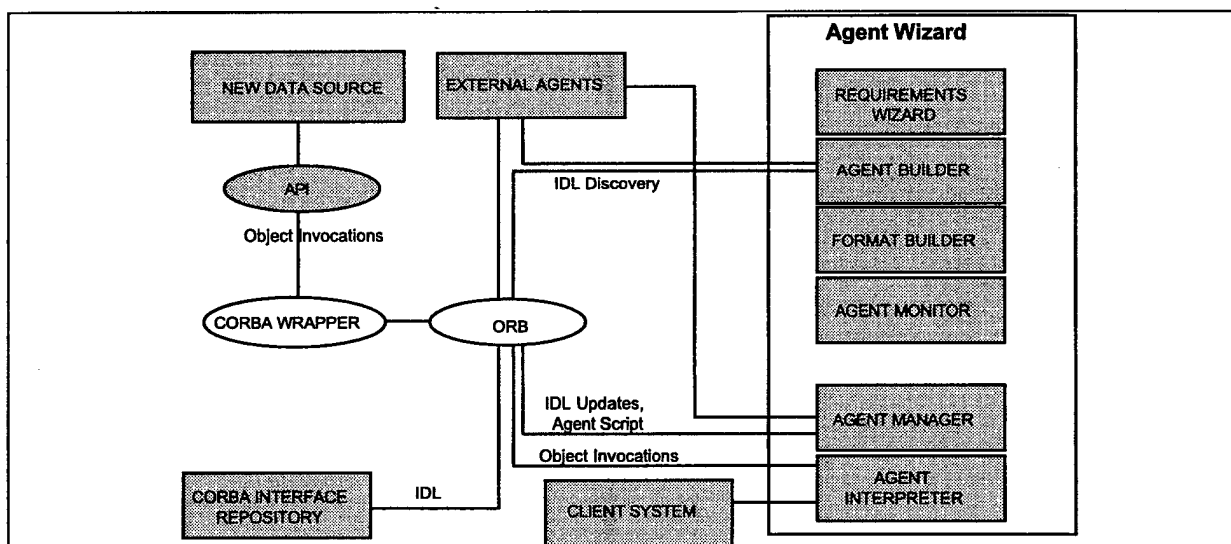
Figure 33.   Agent Wizard in a CORBA World

In the absence of a military infrastructure like DEI, our agents could communicate with data providers on a less advanced CORBA network. In this case, the Gateways would be replaced with data source-specific ORBs and the directory service would be replaced by the Interface Repository (as in Figure 33). There is a commercially available ORB from ObjectSpace, Inc. called Voyager, which could serve a commercial version of the Agent Wizard. Voyager provides the following enhanced CORBA services: name service; directory for cross-architecture object location (RMI/CORBA/COM); gateway for cross-architecture object communication (RMI/CORBA/COM); remote invocation; multicast; security; publish-subscribe for Java events; code mobility (agents that can be autonomous); garbage collection; resource serving through a web server (any HTTP server on the network can be the resource server for classes, and there can be more than one); and object activation (resurrection from permanent storage). All of these features, especially the mobile code support, have led us to believe that Voyager could amply support Agent Wizard-created agents for communicating

with network data sources. If awarded a Phase II contract, we will investigate Voyager's capabilities in this light.

Another commercial possibility is what The Technical Resource Connection (TRC), Inc. calls CORBA Beans. David Houlding of TRC wrote an article in the Sept. 99 Java Report describing this architecture for web-based creation and use of CORBA components for integration into Java programs. This type of architecture could be perfect for application developer customers who have some knowledge of the data servers in question and want to create applications (agents) capable of querying those servers within or in conjunction with their other software. This methodology could also be extended to the general user if an enhanced ORB is used so that less data server knowledge is required by the end-user.

3.2.4.3 Extensible Markup Language (XML)

Although originally designed for a much more comprehensive role, XML data definitions are now being imbedded in web pages for the sole purpose of enabling access to structured data. The Agent Wizard will be able to retrieve data in this architecture but, since the Wizard will have no foreknowledge of the domain or data structures, it will have to rely on the operator's understanding of the data. Figure 34 shows how the Agent Wizard will operate in this environment.
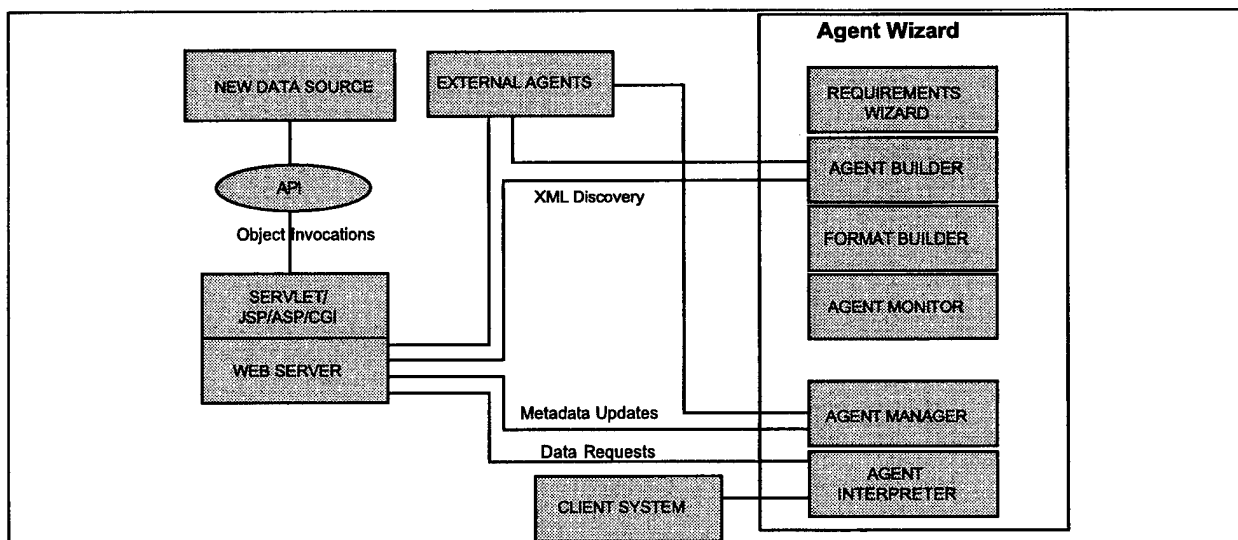
Figure 34. Agent Wizard in an XML World

If neither DEI nor CORBA is available on a commercial network, which is currently true in almost every case, then the Agent Wizard will have to exist in a metadata-sparse environment, like the current Internet (as shown in Figure 34). Right now, an agent can send a Hypertext Transfer Protocol (HTTP) message to a remote server (e.g. www.tvweather.com) containing a request for a specific file or a command to the server to be interpreted by a Common Gateway Interface (CGI) or Active Server Page (ASP) engine. Sending HTTP requests in this way is useful for retrieving known files or querying known data servers for information, but it doesn't lend itself well to dynamic resource discovery, which is automatically provided in DEI and CORBA architectures. There are web server products currently available that provide this type of discovery. One of them is eXcelon's Portal Server, which is capable of responding to a typical HTTP CGI-like query by sending back an XML document describing the data it has available. If this type of product is available on any given web server, an agent could send a general query to the server, get back an XML

43

description of the data provider, and make an intelligent decision as to whether the data provider meets his user's requirements. There are other products currently available for adding XML metadata to web servers, including Bluestone's Visual-XML. One of our tasks in a potential Phase II project would be to survey the existing technologies and develop a demonstration of their usefulness to intelligent software agents.

Similarly, the Agent Wizard will be constructed so as to make it easy for us to clone it for use in other domains, for example financial services, online shopping, etc.

3.2.5 External Funding Search

3.2.5.1 Government Sources

We need to demonstrate that the Phase II project, which would produce a functioning Agent Wizard, will be of direct benefit to existing DoD programs such as DEI and to the overall DoD community. With respect to DEI, there appear to be several options for demonstrating this support.

NRL section head Dr. John Ellis and Ms. Susan Starke have reviewed the project material and have agreed to coordinate DEI with the Phase II Agent Wizard project and assist it in accessing TEDS and other METOC databases via DEI gateways. They have also agreed to take our presentation back to their sponsor and see if they could make some funding available for the following options:

- Phase II Agent Wizard matching funds specifically for the development of DEI related user services that are already part of DEI requirements.

- Phase III funding to assist in the maintenance and continuing development of the Agent Wizard (conditioned on the successful completion of the Phase II Agent Wizard project).

3.2.5.2 Commercial Sources

In addition to supporting military customers, an additional objective of the Small Business Innovative Research program is to create new products and businesses to be useful in the commercial marketplace and to the general public. Thus a goal of our Phase II research program will be to prototype one or more such commercial applications. Our current plan for a commercial application calls for the creation of a web browser plug-in that will automatically access weather data from the World Wide Web when that data would be useful to the user, without requiring any user action.

The commercial application will be a software agent that gathers, analyzes, and displays weather data for users, tailored to their specific needs. It will perform this tailoring using the context of the connected application or web page, modified as necessary by the user or the calling application. It will perform the following:

- By examining data retrieved by the browser, it will know when and what kind of weather information is needed.

- It will seek information from multiple sources, then fuse and store the collected information.

- It will relate information to a specific geographic area or route.

- It will keep the information updated as needed and delete data from the host computer system when it is no longer needed.

We are calling the future commercial version of such agent technology WEATHERDOG (a sample is shown in Figure 35) and we are seeking commercial sponsors and/or partners for this research. We plan to demonstrate an early version of this prototype using specific web pages that are part of major airlines' and travel sites' airline schedule pages. This prototype agent would:

- Detect that an airline schedule page is present;

- Extract airport and date information from the current page;

- Determine whether valid weather forecasts exist for these places and times;

- Fetch these forecasts from one or more providers' sites; and

- Display the information for the user in a new browser window.

In the new window, the user would also be able to "save" a future itinerary, so that the agent would monitor those locations and dates for new or significantly changed weather forecasts and then alert the user appropriately.

Figure 35. WeatherDog Browser Plug-In Display for Travel Weather Forecasts

46

This same concept could be used for many other types of web pages, including any kind of travel or recreation arrangements (skiing, golf, etc.) or outdoor event. It could also be configured so that regular weather reports appear on the desktop for selected locations.

For the early prototype we plan to select one airline and tailor the agent to that particular page, recognizing that page using any unique character string appearing in the HTML. We will then examine the HTML data for dates, times, and locations. For a long-term product, we would expect and encourage providers to begin including XML data descriptions in their pages. For now, we can take advantage of the existing structures in the HTML. Delta Airlines, for instance, conveniently provides the information in data names, as shown below.

```
<INPUT TYPE="hidden" NAME="itn_class_code_0_1" VALUE="K">
<INPUT TYPE="hidden" NAME="itn_dept_name_0_1" VALUE="New York (Kennedy)">
<INPUT TYPE="hidden" NAME="itn_dept_code_0_1" VALUE="JFK">
<INPUT TYPE="hidden" NAME="itn_dept_date_0_1" VALUE="950374800">
<INPUT TYPE="hidden" NAME="itn_dept_time_0_1" VALUE="350P">
<INPUT TYPE="hidden" NAME="itn_dest_name_0_1" VALUE="Boston">
<INPUT TYPE="hidden" NAME="itn_dest_code_0_1" VALUE="BOS">
<INPUT TYPE="hidden" NAME="itn_dest_date_0_1" VALUE="950374800">
<INPUT TYPE="hidden" NAME="itn_dest_time_0_1" VALUE="515P">
```

For the preliminary prototype, we would use just one provider's page as a demonstration. In the Phase II prototype, we would expand this to cover all the major airlines, all the major travel sites, and any site that conforms to a published XML standard. While it is unlikely that Daniel H. Wagner Associates, Inc. will set this XML standard, anything could happen.

Commercial marketing of WEATHERDOG could take more than one avenue. First, a free web browser plug-in could be developed and distributed free of charge. This plug-in could generate revenue from web advertising and click-through fees. Obviously, popular weather web sites would be the first candidates for such advertising fees.

Second, we plan to market an upgraded version with Application Program Interface (API) specifications for use by application developers whose software products could benefit from an organized method of accessing weather data. These applications could include software for commercial sale or proprietary software for internal distribution within a corporation. This "pro" version would have selectable interfaces to popular Geographic Information Systems (GIS) for the purpose of displaying area weather overlays and route coloring in the user's display environment.

During Phase I, we contacted a number of companies as potential funding sources, partners, or customers.

TVWEATHER Mr. Anthony Watts, President of ItWorks, Inc. has been contacted and he is definitely interested in considering all levels of cooperation and funding. ItWorks runs a number of sites that may have real needs for automatic agents.

- www.itworks.com
- www.tvweather.com
- www.weathershop.com
- www.spacewarn.com
- www.nowcaster.com
- www.weatherwarn2000.com

DotCom Stores, Inc., a Hampton based internet incubator, is seriously considering a strong entry into the internet travel business. President Mark Hollingsworth has already verbally agreed to include a version of WEATHERDOG as a proprietary web-based feature of this site for a limited period in exchange for later use of their portal for downloading the browser plug-in version.

Praxair, Inc. We have discussed the planned features of WEATHERDOG with this industrial gas company with a global distribution system, who are developing their own internal route planning system. The addition of weather information to route displays would assist schedulers in deciding which sites to stock up with gas supplies in advance of predicted bad weather. This firm has already expressed an interest in sponsoring this research in exchange for future internal product licensing rights. Our Agent Wizard would be an excellent candidate to provide required weather access to their development, at very low cost. We plan to visit the Praxair headquarters in the near future.

Weather.com is a subsidiary of the Weather Channel, Inc. Although we signed a non-disclosure and visited their headquarters in Atlanta, they decided not to invest in the Phase III development, because of their focus on the general public as opposed to the smaller market of the business traveler. They thought that the utility of WEATHERDOG was limited to that market.

IWS is another subsidiary of the Weather Channel, Inc. and the Weather.com president suggested a contact person there who might be more interested than they. We have not been able to make contact there so far.

3.2.6   Patent Filing

We have submitted a preliminary patent application for the context-sensitive data retrieval agent concept (e.g., WEATHERDOG) and are considering other patent options in connection with the Agent Wizard.

4.   Phase II Plans

In Phase II, we plan to create the first fully-functioning version of the weather-oriented Agent Wizard. We expect that product to be immediately adaptable to the work we are doing to create a standard weather retrieval agent for the U.S. Navy mission planning community. That program is now being supported by multiple Navy agencies in a Phase III project. This Phase III project is described in Section 4.1.

4.1   Navy Phase III Technical Objectives

The specific objective of the Phase III project is to provide Meteorological and Oceanographic (METOC) data to all mission-planning functions in the Joint Mission Planning System (JMPS) and to other mission planning systems such as TTWCS through a single software interface called METPLAN.

METPLAN will exist as either a JMPS Framework component fulfilling a requirement to supply METOC data to all JMPS UPCs and/or as a stand alone GCCS segment for use by any non-JMPS MPM segment. It will be DII COE compliant, so that it can be accessed and used by any other DII COE-compliant system on the GCCS LAN.

METPLAN will automatically fetch the most recent and relevant weather products and numerical data from TEDS. It will supply those data to the client MPM, based on a one-time

48

registration message from the client, specifying the desired interface and weapon-specific data format. The client MPM can incorporate those data in mission planning, as appropriate.

In addition to data handling and safeguarding, METPLAN will continue to provide graphical displays of the effects of the weather data on the weapon of interest and displays of the METOC products in the mission area. This feature will require no modification of JMPS UPCs, other than to send the initial registration message, because the JMPS architecture will permit METPLAN to display graphics objects on the tactical window independently. The METPLAN Preferences and Control menus are also independent of the UPC. This feature will enable developers of all JMPS UPCs to integrate METOC data into their function at minimum cost and in a uniform fashion. It will also provide a single display style for the METOC effects and products, across all weapon types.

The METPLAN approach and philosophy is to provide a single form and method for identifying UPC METOC requirements, identifying the data and products that will meet those requirements, and single method of providing a conduit for the selected data into the UPC. The current METPLAN software uses the Preferences GUI to allow the Mission Planner to select a weapon and set the limits for cautionary displays and effects. This GUI also provides a list of catalogued data and products that will be extracted from the TEDS database and made available to the UPC.

The current version of METPLAN runs in conjunction with PFPS as a single-user application. JMPS Ver 1.0 includes PFPS functionality, and we understand that JMPS 2.0 (the first version to include weapon-specific UPCs) will not only keep the PFPS functionality, but will preserve the key elements we need to run METPLAN as an independent agent.

4.2   Agent Wizard Phase II

4.2.1   Technical Objectives

The goals of the Phase II effort will be to create a working prototype of the Agent Wizard and apply that prototype to the development and enhancement of the Navy's METPLAN mission planning weather agent. The specific objectives are:

Objective 1:   Build Prototype Agent Wizard. To prove the concept of the Agent Wizard, we will need to construct working code that performs all the functions required of the Wizard.

Objective 2:   Build DEI Coordination and Cooperation. To insure that the Wizard will work smoothly with future access methods, we will need to insure that the development is in lock step with the DEI progress of the Navy Research Laboratory.

Objective 3:   Expand Schemas to Other Sources. In order for the Wizard to be truly universal, it should have knowledge of schemas that allow it access to all important environmental data sources.

Objective 4:   Expand the Wizard to Include JavaBean and CORBA Compliance. For commercial viability, the Wizard will need to be able to create JavaBean- and CORBA-compliant agents and incorporate the Wizard into an Integrated Development Environment.

Objective 5:   Define the Agent Wizard's Interaction with Other DARPA Projects. In order to keep on the cutting edge of agent technologies, we will coordinate our Agent Wizard work with DARPA's other agent-related projects.

49

Objective 6: Extend the Wizard into Other Knowledge Domains. For the final proof of concept, we will need to extend the Wizard into one or more additional knowledge domains.

4.2.2 Objective 1: Prototype Wizard

For this objective, we will need to achieve a number of milestones. We will need to define the METOC vocabulary used by agents (based on DEI's data model). We will define the translation mechanisms needed, including metadata translatations (DEI to XML) code translations (XML to Java/Perl). We will next have to design and develop Agent Wizard components. The most important of these are:

Requirements Wizard: maps user needs to internal data structers or XML data stream.

Agent Builder: maps DEI providers and requirements definition to agent actions and Java/Perl code.

Agent Manager/Monitor: provides user with the means to control and monitor agent behavior.

Agent Interpreter: executes the internal Agent Language Script agents.

-- In military, invokes the proper DEI gateway methods

-- In commercial world, invokes the proper CORBA object or JavaBean/CORBA Bean methods

Format Builder: provides user with ability to define XSL style sheet for formatting retreived data.

We will then create agent templates for specific end-user profiles (METOC officers, aircraft mission planners, missile mission planners, search and rescue planners, etc.), to complete a military-style product prototype.

4.2.3 Objective 2: DEI Coordination

We will coordinate with NRL-Stennis (Susan Starke and John Ellis) as they continue to develop DEI; they are developing an enhanced CORBA wrapper around TEDS, and eventually around every METOC data source, then other types of data sources in the Navy, and then other services. Their schedule is to have the TEDS version wrapped up in September 2000 and to have the full DEI specification January 2001.

January 2002 will be their full system stand-up, which fits perfectly with the tail end of a two-year Phase II project. This step will include document/design review, phone and face-to-face meetings with DEI personnel in Mississippi, and Agent Wizard design review based on DEI findings.

Incorporate Agent Wizard into a DEI-enabled environment, which will include TEDS. This involves programming the agent code toward DEI specification and testing of Agent Wizard's ability to create agents toward the DEI architecture.

4.2.4 Objective 3: Expand Schemas

To achieve this objective, we will extend agents to other resources (web sites, databases) in the METOC domain. This involves writing DEI wrappers (or getting DEI to write new wrappers) for the NRL-Monterey COAMPS web site, the Point Mugu Naval Weapons Center Meteorological Detachment web site, the Master Environmental Library (MEL), and other web sites and databases that become available during the development period.

4.2.5 Objective 4: JavaBean and CORBA Compliance

To satisfy this objective, we will incorporate agent creation process into a JavaBean-compliant Integrated Development Environment (IDE) by programming agents toward JavaBean specification and integrating the Agent Wizard interface into the IDE.

4.2.6 Objective 5: Define the Agent Wizard's Interaction with Other DARPA Projects

DARPA is currently working on the Control of Agent-Based Systems (CoABS) project, the goal of which is to provide a framework (called the "grid") within which independent agents can communicate and collaborate to achieve their individual and team-oriented goals. [16] This project will be closely related to another project mentioned earlier in this report, the DARPA Agent Markup Language (DAML). DAML's goal is to provide a language through which agents can communicate their ontologies to share information and drive each other's behavior. A major part of our Phase II work would be to determine the Agent Wizard's place on the grid, determine the agents' understanding and use of DAML, and develop a demonstration of Agent Wizard on a CoABS grid with DAML-enabled agents.

We will also explore another DARPA project, Active Templates, the goal of which is to provide easy-to-use spreadsheet-like templates for end-users to quickly perform routine tasks and define new tasks based on their level of expertise in any given knowledge domain. We envision a close relationship between tasks defined on the template and agents created or invoked behind the scenes to accomplish those tasks, very much like the Requirements Wizard leads to agent definition and creation. Phase II work would be to determine how the end-user may drive agent behavior through his interaction with the Active Template program.

4.2.7 Objective 6: Other Knowledge Domains

To achieve this objective, we will need to extend Agent Wizard's capabilities to one or more other knowledge domains, as identified during the Phase II work and subsequent demonstrations to end-users and meetings with knowledge domain experts.

We believe that all these objectives can be fulfilled within the time and funding constraints of a Phase II SBIR award.

References

[1]   Report on Information Management to Support the Warrior, United States Air Force
        Scientific Advisory Board, SAB-TR-98-02, December 1998.

[2]   The Dublin Core: A Simple Content Description Model for Electronic Resources,
        Dublin Core Metadata Initiative, http://purl.org/metadata/dublin_core, August, 1999.

[3]   The Global Information Locator Service, http://www.usgs.gov/gils, August, 1999.

[4]   Reggie – The Metadata Editor, Distributed Systems Technology Centre,
        http://metadata.net/dstc, December, 1998.

[5]   DSTC HotMeta, Distributed Systems Technology Centre,
        http://www.dstc.edu.au/Products/Resource_Discovery/HotMeta/hotmeta_intro.html,
        March, 1999.

[6]   Instant CORBA, p. 7, Robert Orfali, Dan Harkey, Jeri Edwards, Wiley Computer
        Publishing, 1997.

[7]   DSTC HotOil, Distributed Systems Technology Centre,
        http://www.dstc.edu.au/Products/Resource_Discovery/HotOil/hotoil_intro.html, March,
        1999.

[8]   The SHOE FAQ, Parallel Understanding Systems Group, Department of Computer
        Science, University of Maryland at College Park,
        http://www.cs.umd.edu/projects/plus/SHOE/faq.html.

[9]   DAML.org, DARPA Agent Markup Language (DAML) Public Portal,
        http://www.daml.org.

[10]   Overton, R.H., "Tomahawk Land Attack Missile (TLAM) Environmental Factors,"
        Daniel H. Wagner Associates report to Naval Research Laboratory, Dec. 1992.

[11]   Giannola, R.M., "Implementation of METOC Data in TTWCS for Launch Platform
        Mission Planning," Draft White Paper, Oct. 8, 1999.

[12]   Rosmond, T.E., "The Design and Testing of the Navy Operational Global Atmospheric
        Prediction System," report to Office of Naval Research, June 1992.

[13]   Errico, R.M., Rosmond, T.E., and Goerss, J.S., "A Comparison of Analysis and
        Initialization Increments in an Operational Data-Assimilation System," Monthly
        Weather Review, Vol. 121, pp. 579-588, Feb. 1993.

[14]   Hodur, R.M., "Coupled Air-Ocean Mesoscale Modeling at the Naval Research
        Laboratory," Brief by Marine Meteorology Division, Naval Research Laboratory.

[15]   Hodur, R.M., "Development and Testing of the Coupled Ocean/Atmosphere Mesoscale
        Prediction System (COAMPS)", report to Office of Naval Research, Nov. 1993.

[16]   Control of Agent Based Systems (CoABS) Public Portal,
        http://coabs.globalinfotek.com.