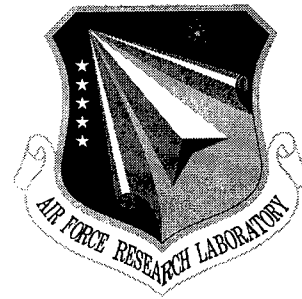AFRL-IF-RS-TR-2001-32
Final Technical Report
March 2001

INTELLIGENT ACCESS TO LARGE KNOWLEDGE BASES FROM HETEROGENEOUS DEVICES USING MULTIPLE PROTOCOLS

Capraro Technologies, Inc.

Gerard T. Capraro and Gerald B. Berden

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

20010507 072

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

Although this report references (*) limited technical reports on page 21, no limited information has been extracted.

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.
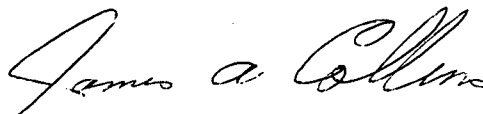
AFRL-IF-RS-TR-2001-32 has been reviewed and is approved for publication.

APPROVED:

JOHN SPINA
Project Engineer

FOR THE DIRECTOR:

JAMES A. COLLINS, Acting Chief
Information Technology Division
Information Directorate

REPORT DOCUMENTATION PAGE		*Form Approved* OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE MARCH 2001	3. REPORT TYPE AND DATES COVERED Final Apr 98 - Dec 00	

4. TITLE AND SUBTITLE
INTELLIGENT ACCESS TO LARGE KNOWLEDGE BASES FROM HETEROGENEOUS DEVICES USING MULTIPLE PROTOCOLS

5. FUNDING NUMBERS
C - F30602-98-C-0171
PE - 62702F
PR - 5581
TA - 27
WU - PO

6. AUTHOR(S)
Gerard T. Capraro and Gerald B. Berden

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Capraro Technologies, Inc.
311 Turner Street, Suite 410
Utica NY 13501

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory/IFTD
525 Brooks Road
Rome NY 13441-4514

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

AFRL-IF-RS-TR-2001-32

11. SUPPLEMENTARY NOTES
Air Force Research Laboratory Project Engineer: John Spina/IFTD/(315) 330-4032

12a. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
This report documents the results of an effort with an objective to demonstrate the feasibility of integrating Artificial Intelligence (AI) technology with web technology to bring very large data and knowledge bases to a hand-held computing device. A software architecture is provided consisting of three levels of intelligent software assistance, i.e. a personal assistant, a hardware assistant, and a structured query language (SAL) assistant. To demonstrate the software architecture a USAF Air Mobility Command operational problem domain is simulated to represent the large database from which military personnel want to gather information. A brief overview of the USAF Scientific Advisory Board's Joint Battlespace Infosphere (JBI) is presented with references of how the resultant software developed here instantiated portions of the JBI architecture. The software recognizes a user profile including their computer device and tailors the presentation of information the the user accordingly. the software architecture provides push and pull paradigms. The user may change their profile at any time and the system appropriately responds. A description and demonstration of the software are provided that show how one can access data via an http connection and through email.

14. SUBJECT TERMS
Knowledge Bases, Databases, JBI, Artificial Intelligence, Air Mobility Command

15. NUMBER OF PAGES
80

16. PRICE CODE

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

Table of Contents

List of Figures

1.0 Introduction

The Information (IF) Directorate of the USAF Research Laboratory has been investigating the integration of large databases and knowledge bases for more than ten years. Personnel at Capraro Technologies, Inc. (CTI) have worked on one of these first efforts (5 - 9) during the 1980's. Since then, the IF Directorate has continued to pursue numerous efforts by many experienced researchers. A next step in this process is to provide simple access to this wealth of data and information from anywhere in the world.

Military personnel need to communicate with superiors and command centers. Over the years these communications have been primarily audio. However, with the advances in computing and communications it is now possible to communicate to anyone with a lap top computer, a consumer electronics (CE) device, or a personal digital assistant (PDA), given a phone line or a radio frequency (RF) modem. One can stay in touch with his/her e-mail, send or receive faxes, access applications on a home computer, and query knowledge and databases anywhere in the world. He/she can have access to very large amounts of data and information in any form (i.e. voice, graphics, and video). It was previously shown feasible by CTI (13) that multiple databases could be accessed over the web and presented to a hand-held computing device (HCD) using telephone and cellular phone connections.

The objective of this effort is to demonstrate the feasibility of integrating Artificial Intelligence (AI) technology with web technology to bring very large data and knowledge to a HCD in an efficient manner. The first interim report provided an overview of the relevant technologies required to meet this effort's objective and can be found in Appendix A. The second and third interim reports provided the evolution of our software and hardware architecture design. Section 2 of this final report provides a short background for this effort. Section 3 provides a brief overview of the Department of Defense's Joint Battlespace Infosphere: a new distributed information system. Section 4 describes the USAF problem domain chosen to demonstrate our approach for bringing information to a user. Section 5 describes our current software architecture. Section 6 describes the demonstration we have developed applying our software architecture to the defined problem domain. Section 7 provides a summary of our work along with our conclusions and recommended future work. Appendix A documents the results of a literature review of relevant technologies and the state of hand held device technology.

2.0 Background

Legacy databases exist throughout the US Air Force. These databases were costly to develop and are still costly to maintain. They are the backbone to many military decision processes utilized throughout the US Air Force, during both peace and conflict. Research pursued by the Air Force has been in an effort to integrate these databases in order to allow individuals to query them as if these databases were all on the same machine being maintained by the same database management system (DBMS). This capability allows the individual databases to continue to be

developed and maintained in a consistent manner and yet allow many users to integrate their data with other databases and reap the benefits of their synergy.

The development of browser technology, the Internet, and intranet systems now provide us the capability of integrating these disparate databases with a common user interface. This capability is independent of the DBMS and, to a certain extent, its resident machine. This allows a user to interact with these integrated databases over the Internet as if they were a single homogeneous database. Some major DBMSs provide the capability of accessing databases over the Internet using browser technology. However, they are DBMS dependent and do not allow for the integration of numerous heterogeneous databases hosted by different DBMSs.

In our last effort (13) we conjectured that the integrating of very large data and knowledge bases using web technology is very feasible and that to make it available to HCDs was possible but required intelligent software running on the web. The intelligence is necessary because the HCDs are varied and limited in capability and their mode of connection to the web are variable and bandwidth limited. A potential future implementation that was proposed is presented in Figure 1. Here we showed how a personal assistant would exist for each user, how that assistant could call upon intelligent domain agents and also described the seamless accessibility to heterogeneous databases located anywhere using DARPA's I3 technology. We believe that this architecture has merit and portions of it can address our concerns of providing access to heterogeneous databases on the web via HCDs.
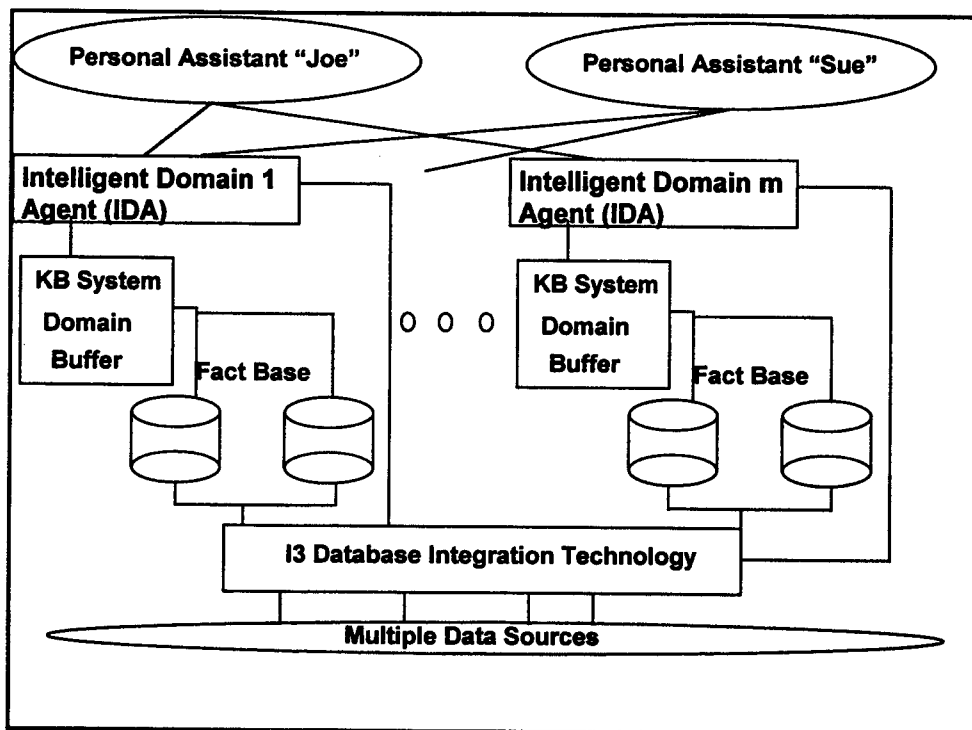

Figure 1. An Intelligent Preliminary Architecture

2

In this previous effort, we investigated Java as the language for building the user front-end and accessing different DBMSs utilizing Java's database connectivity (JDBC) capability. Java programs run on a Java virtual machine within a browser or as a native installation. We successfully implemented this platform independent capability as a proof of feasibility demonstration by hosting the user interface software on a server and down loading it to a HCD within an HTML form. There are numerous issues that must be investigated in order for the system to provide useful data to a HCD as compared to a PC, e.g. processing speed, programmability, memory, screen size, and available bandwidth. Part of our objective in this effort was to investigate and demonstrate ways to overcome these limitations in order to retrieve and input data from a HCD as easily as one can from a PC.

During the third phase of this effort it was brought to our attention that there existed a United States Air Force Scientific Advisory Board (USAFSAB) "Report on Information Management to Support the Warrior". We reviewed the available material (14) along with subsequent documents related to the Joint Battlespace Infosphere (15) and concluded that there are major intersections and goals with our effort and what they have proposed in their reports. We are only demonstrating a portion of their design and will highlight those portions by using their terminology where appropriate in the following sections.

3.0 Joint Battlespace Infosphere

The Joint Battlespace Infosphere (JBI) is a Department of Defense (DOD) information management system (14,15). The following is a brief overview of the system obtained from the references. The result of our effort addresses some of the core issues of a JBI providing information to a user.

The JBI integrates and assembles data from multiple sources and distributes resultant information in the proper form to the appropriate level of personnel. The JBI is built upon a collection of protocols, processes, and core functions that allow for the sharing and exchanging of information. The JBI attempts to integrate legacy "stovepipe" information systems by acting as an intermediary between them so that they can share consistent information. In addition by acting as an intermediary between systems it attempts to enhance the general pool of information through synergistic efforts between the individual systems' information. It also filters and presents information relative to the individual user's profile and needs.

The JBI architecture is based broadly upon four key concepts along with numerous supporting technologies. The four concepts are:

1. Information exchange via a publish and subscribe paradigm,
2. Data are transformed into knowledge via fuselets,
3. Collaboration between distributed clients is via updateable knowledge objects, and

4. Defined force templates are used to incorporate military units into the joint task force.

Some of the supporting technologies are Browsing, Interaction, Fusion, Objects, Structured Common Representation, Automatic Data Capture, and Tailoring Information To Meet User Needs. It is this last technology that our effort addresses most effectively. In the USAFSAB report (14) it states that: "The understanding of a situation or the available options depends critically on presenting the information in an appropriate form." ... "the presentation must be tailored to the workflow task and to the preferences of a particular user. What is presented in the cockpit may be very different from what is presented in a command center." As a specific example: "When a platoon of ground troops requests the location of enemy tanks, the JBI provides that information in a form tailored for the personal digital assistant carried in the field."

The goals of our effort are but a small subset of the JBI goals. Our objective was to demonstrate the feasibility of integrating AI technology with web technology to bring information in an efficient manner to a HCD. In order to show the value of our work it was suggested that we demonstrate this capability using a well defined USAF problem domain. While performing this effort the JBI report was released. The timing was very fortuitous, so we added some of our own resources to expand and tailor our demonstration to showcase solutions to some of the JBI goals.

4.0 Problem Domain

To demonstrate the approach and its benefits we have chosen a particular need within the USAF as obtained from Air Mobility Command (AMC). We are currently performing as a subcontractor to Litton TASC on the Information For Global Reach (IFGR) contract with the IF Directorate of the USAF/RL. In that role we have become familiar with AMC's need to track flights throughout the world: where flights are located, what crew is on board, what are their current flight paths, etc. In that light we have created multiple database relations that capture some of AMC's database that would be populated from messages through IFGR. Once the messages are stored within a database management system, then it is our goal to demonstrate how different persons can retrieve information regarding this data regardless of available bandwidth or processing devices.

Within the AMC problem domain, aircraft will send messages to the ground and be entered into a database for every flight. A subset of those messages would be sent when an aircraft departs a location, an auto position report/message is generated at preset intervals in flight, a message would be sent containing the time when the aircraft touches down and when a flight arrives at the gate. We also obtained a database of the International Civil Aviation Organization (ICAO) codes. This database contains the latitude, longitude, name, country, its ICAO code, and elevation in meters for 5,760 airports around the world. We generated a hypothetical database for passengers, their addresses, and their ranks. We also generated database relations with hypothetical crewmembers for different flights. We elected to generate these hypothetical data rather than use more realistic data for obvious security reasons.

4

In order to demonstrate a user acquiring data from an actual database in real-time we needed to populate a database in "real-time". To do this, we wrote a program to calculate a flight's great circle path. It allows a user to choose a departure and arrival ICAO airport codes, departure time, altitude and speed. The program will then generate the great circle path for the flight. We generated five days worth of flight scenarios with five flights per day. Each flight contains a crew and a random number of passengers.

To provide a visual display of where these flights are located, we developed another program that approximates the latitude and longitude on a map to display the locations of the aircraft during flight. The map can be modified with colored diamond shaped symbols depicting the location of an aircraft when they last sent a message. This depiction of the position of the flights is suited for displaying where one or more aircraft are located at any one time. However, it does not provide a time history for all flights during a day. To address this issue we generated another graphic that provides a time line for the current status of each aircraft or flight number. Here we represent the time that each flight sent a message regarding its departure, auto position, and landing. These different graphics and processes will be discussed again in more detail in Section 6.

5.0 Software Architecture

The foundation of our software architecture is the integration of numerous heterogeneous databases using DARPA's I3 technology. For this effort we are emulating this capability by using a relational DBMS which contains both the application database and its meta-data. For modeling purposes we are assuming that all interfacing to the database and its description will be done in a Structured Query Language (SQL). We currently envision three levels of rule based AI functions to be performed upon these data. These are an SQL Assistant, Hardware Assistant, and a Personal Assistant. See Figure 2. All of the facts or data required by these assistants will be stored within the DBMS that is maintaining the application databases. The rules within each of these assistants will be kept within their intelligent objects.
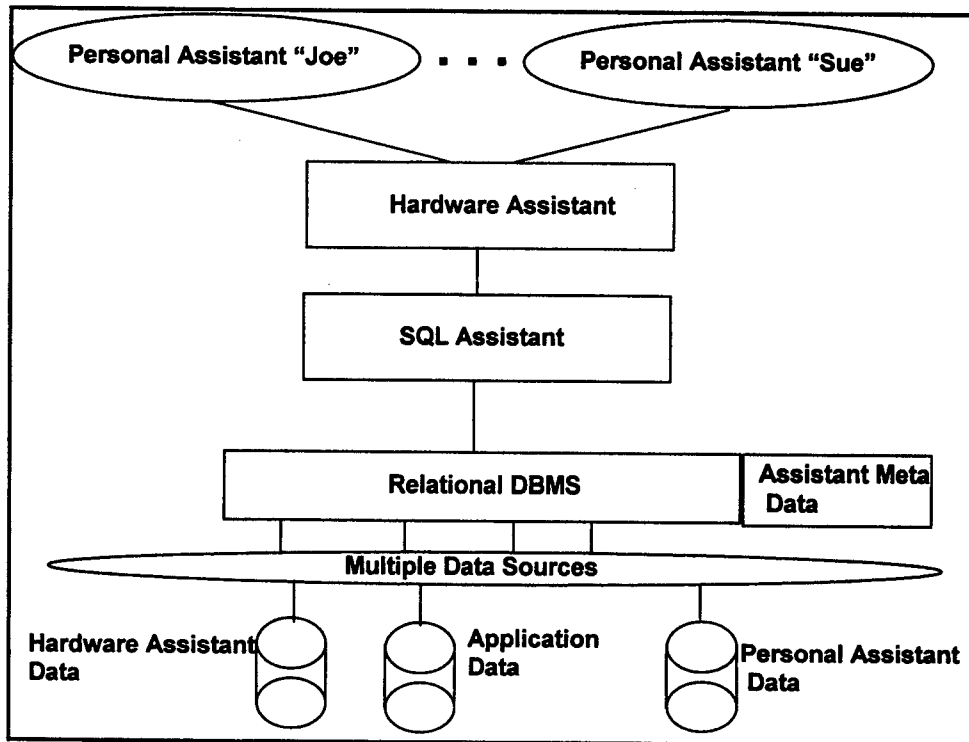
Figure 2 Prototype Architecture

SQL Assistant

The SQL Assistant will allow us to evaluate the different functions that are performed against the data, whether they are a simple query, massive update to the database, or simply providing an ad hoc query capability. The assistant will allow us to provide and maintain additional meta-data to the database that will be needed by the Hardware and Personal Assistants.

Consider the following SQL statement:

Select *
From **tblMission**
Where **MissionPriority** = **"High"**

The first statement requests all the attributes contained within the "From" statement. The next statement says that the relation is tblMission and the "Where" statement says we should select only those occurrences which have a "High" mission priority. The SQL assistant will interpret this statement and determine how many attributes are contained in tblMission, how many characters are required to display them all, whether all the attributes are text, or some are integers, or some are images, etc. This information will be passed along to the Hardware Assistant and Personal Assistant for making decisions on how to partition the processing

6

between the server and the terminal device, how to package the result, and communicate with the user.

These statements will allow a designer to add the additional meta-data related to each of the attributes. For example, whether they should be sorted, processed into statistics, counted, duplicates removed, or used in performing time line analyses.

Hardware Assistant

This assistant allows the designer to add device-specific facts to the meta-database. These include bandwidth connections to devices, device processing capabilities, software processing capabilities, audio, video, screen capabilities, etc. These facts (data) will be stored for each device that can possibly have access to the integrated databases. These facts, working along with the SQL Assistant, will allow the Hardware Assistant to decide with the Personal Assistant how best to partition the processing and send the information to the device, e.g. choose the proper HTML page, send Java code, or formulate an email containing the response to the query in a text message.

For the example above, the Hardware Assistant will determine the size of the response. Based upon the user and the HCD, it will decide whether it should send the total response, or perhaps just five percent (with a user "drilling" down to obtain the information they require), or send a group of statistics describing the response (such as the number of occurrences, and the first, last, and middle occurrences based on date). The possible responses will be domain, user, and device dependent.

Personal Assistant

This assistant maintains information about individual users. There is an assistant for each unique user. Facts are stored in the database that relate to each user, and include email addresses, personal pages, security codes, standard tools and formats, etc. It is this assistant that communicates with the user and allows he/she to tailor responses to meet his/her needs. These facts are defined by the designer and can be tailored to a certain extent by the user. Each connection by a user to the server will be controlled by their Personal Assistant.

For the example above, dependent on the response to the SQL request, the Personal Assistant can notify the user that the response will take X minutes to send to their HCD. It will then ask the user if he/she would like to receive the "raw" data as email, and then send summary information to the HCD. It could also suggest other statistics that could be provided, and suggest other ways to display results without sending all of the data. Since the interactions with the user occur over a period of time, the Personal Assistant will learn how each user likes to view results, when accessing information with a particular device.

The architecture provided in this section is not complete but ongoing. A portion of the architecture was built and will be discussed by describing a demonstration that was given and hosted on our web site for the USAF to use for future demonstrations and evaluation.

6.0 Demonstration

In the previous two sections we described the problem domain and a generic software architecture solution that we are investigating. This section will describe a demonstration of applying the software architecture to the AMC problem domain. Although we have talked mainly about HCDs, the architecture must also serve numerous devices depending upon the user and their profile. The types of devices we currently envision are shown in figure 3.

A full workstation will normally have a large screen with a browser interface and Java capability. The bandwidth available and modem may be an issue and should be considered for example, when dealing with large images, audio, and video. On the other end of the spectrum are Palm OS machines and cellular telephones with minimal screen sizes and low bandwidth. However, they can send and receive email messages using a telephone connection, a Radio Frequency (RF) service or through their wireless phone. These devices are evolving rapidly and are especially popular outside the US. In the middle lies those devices that have some browser capability, low bandwidth, and may or may not be Java enabled. Typical classes of devices are those with Microsoft Windows CE operating systems. Since their display screens are small, they are hampered in browsing many of today's web sites.
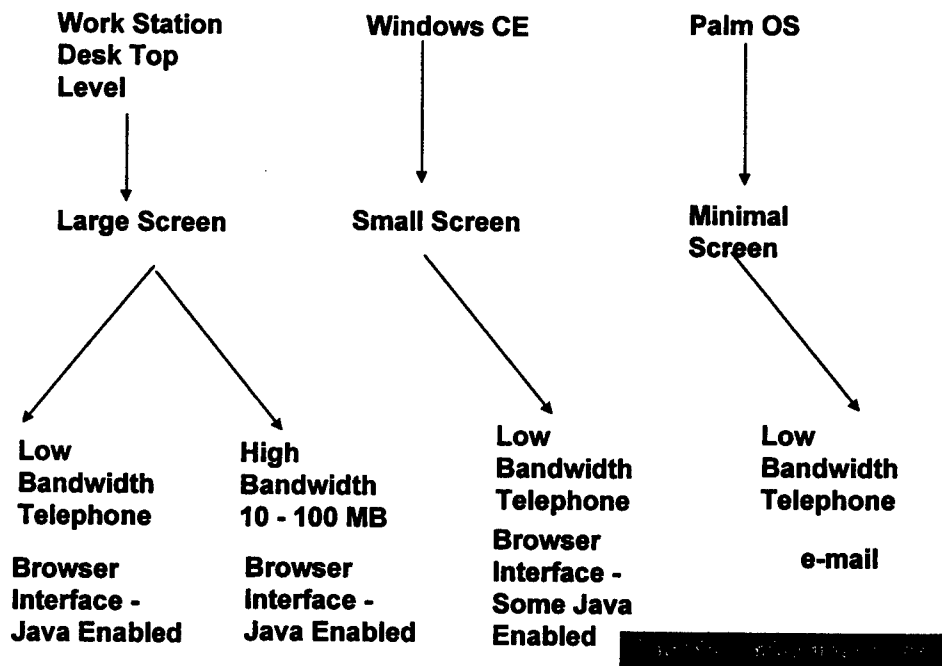
Figure 3 Types of Devices

In our demonstration we wanted to exercise some of the major aspects of the JBI. We wanted to demonstrate both push and pull of information depending upon a user's profile and the dynamics of the data. We also wanted to demonstrate in some small way the publish and subscribe paradigm. We also wanted to show that we were dynamic in our capability to change processes and control mechanisms in real-time. Lastly, we wished to show a use of fuslets as defined within the JBI reports. Noting that we could not use the actual data that are contained within the AMC database we needed to create a capability that would simulate the messages that would be generated within their system.

The simulation code that we generated was based upon a great circle path that an aircraft might fly from one location to another. We found a simple algorithm on the web and implemented it in Java. We also found a list of all the ICAO codes with their respective latitudes and longitudes. With these two resources in hand we were able to compute a path given the speed of an aircraft and its altitude. We also needed a way of describing to a user where aircraft were located at any point in time. We developed a simple method using a world map which we partitioned into sections and approximated an aircraft's position on the map using a linear approximation depending upon the section it was located. This approach is not highly accurate but it suited our purposes for displaying the relative locations of all the aircraft at any point in time and we could provide the capability in Java within the resources available.

The user creates the flights by choosing the day, up to five flights, assigning a mission number for each flight and selecting the ICAO code of departure and destination. Given these data and the aircraft's speed and altitude, the program will then generate the departure message, the 15-minute position messages in route, and the landing and arrival gate message. A random number generator was developed to assign the additional time that it would take the aircraft to reach the gate once it lands. We generated five flights of messages for each of five consecutive days. The data have been pre computed and stored within a relational database. For demonstration purposes once the user chooses which day they wish to simulate, another Java code searches the database and generates the messages and stores them into another database as if they were messages received from an actual aircraft for the day chosen. A user acquires this opportunity when they first enter the demonstration web page (see figure 4).

9
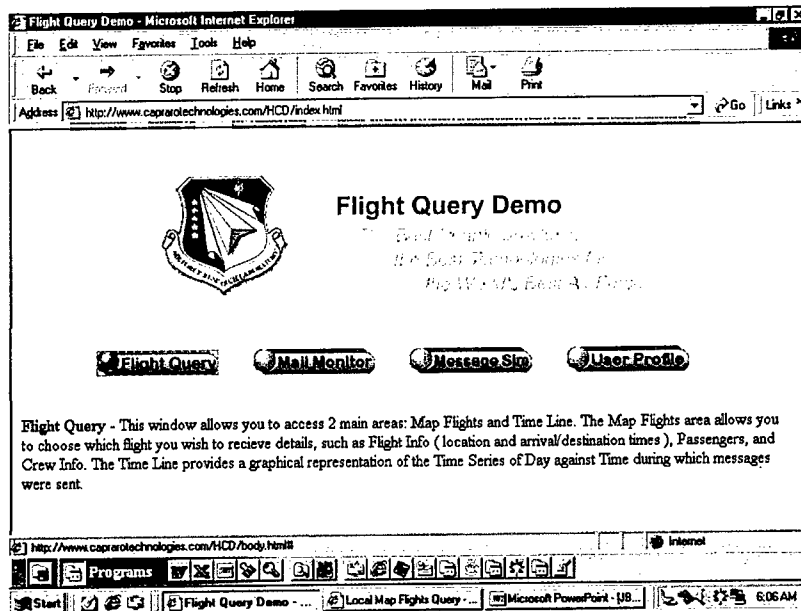
Figure 4. Demonstration Switchboard Page

Once the user enters the switchboard page they can maneuver around the demonstration pages and functions. When the cursor is over each of the activation buttons a text description is displayed in the lower third of the page. In the case above, the cursor is over the Flight Query button. For moving to the message simulator page the user would click on the third button from the right, labeled Message Sim. Once activated, a page similar to the following will appear.
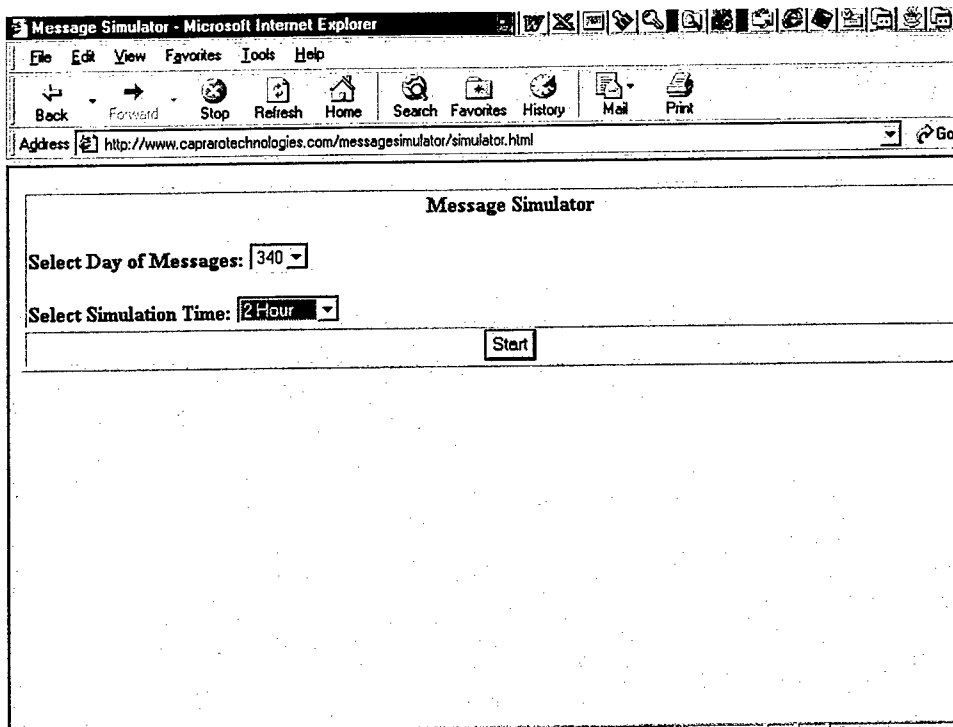
Figure 5. Simulator Page

The user can choose one of five Julian days to simulate, numbered 340 through 344. Next, the user can choose whether to run the simulation for a total of 24 hours, over a simulated period of 0, 1, 2, 3 or 4 hours. The first choice of 0 hours was provided for diagnostic checking of our software. If the user wants to provide the demonstration over a short period of time they can choose the 1-hour option.

The demonstration displays results based upon a user's profile that can be changed in real-time, meaning that the system will respond to the user's change in profile immediately after the change is recorded within the meta-database, as described earlier. The user profile page is accessed from the main switchboard (figure 4) by activating the button on the far right. Once activated, a current list of all the users is shown and the capability of adding a new user is provided. The current design only allows a user to enter name, email address, and type of device (see figure 6). The other notification entities and rank are currently not functional. This information is stored in the user profile meta-database. The types of devices are either email-only devices (such as a Palm), or PC or workstation type computers that are web enabled. It is this information the system will use to push the information to the user, or will determine how a user will pull from the system.

11
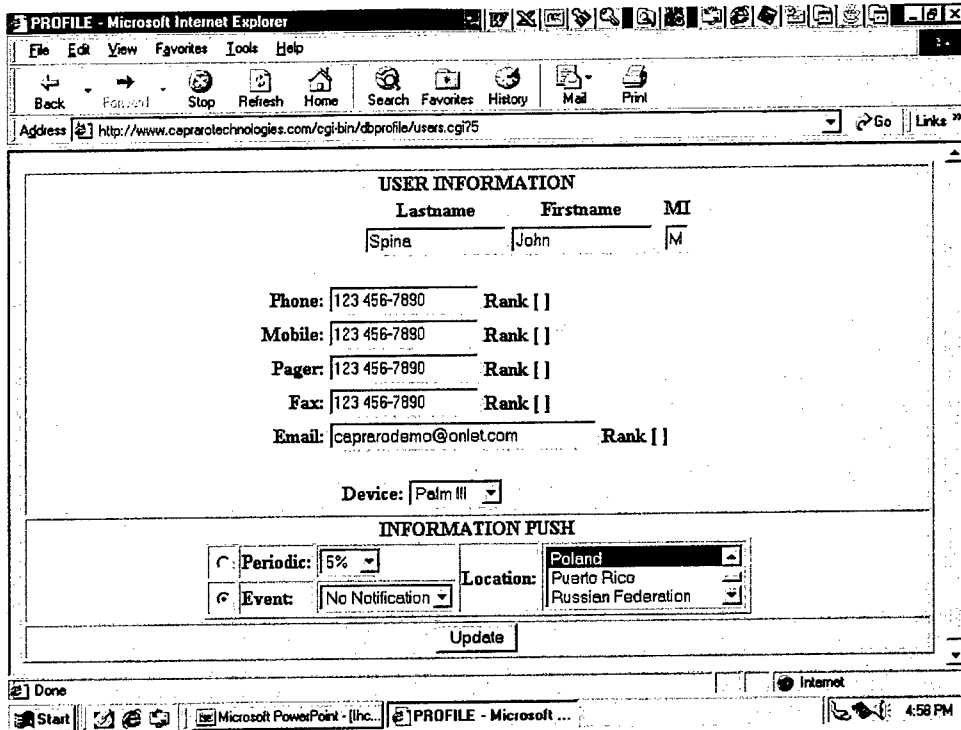
Figure 6. User Profile Page

The lower portion of figure 6 allows the user to define how information is pushed to them via email. The user can choose whether they wish to have the status of flights provided to them periodically or only when triggered by specific events. In a real system the periodic choices would be based upon hours or minutes. Since this is a simulation generated system, the user can choose percentages of the time for the total day's simulation. For example, if the user chose 5% and the total simulation time chosen (as shown in figure 5) was one hour, the user would receive an email every 3 minutes for that hour. A sample email message is shown in figure 7. If the user chose to have their email messages sent to them triggered by an event, then the two options now available are either a departure or an arrival from a particular country. For example if a user chose Event equal to departure and Location equal to United States, then any time an aircraft departed from the US, a message status of all flights for that day would be sent, similar to the message shown in figure 7. The user can also request that no messages be sent to them (Event equal to No Notification) and receive flight status information by pulling the data from the site themselves.

Flight Number: MC0503
Departure Time: 1630
Departure Location: K63G (Chicago / Calumet Coast Guard Station)
Destination Location: KLAX (Los Angeles, Los Angeles International Airport)
Last Reported Time: 1856

Flight Arrived at: 1856
Reached Terminal at: 1910

Flight Number: MC0603
Departure Time: 1045
Departure Location: EDDG (Muenster / Osnabrueck)
Destination Location: TISX (Christiansted / Alex. Hamilton Field, Saint Croix)
Last Reported Time: 1653
Flight Arrived at: 1653
Reached Terminal at: 1700

Flight Number: MC0703
Departure Time: 1755
Departure Location: EGXH (Honington Royal Air Force Base)
Destination Location: KMXF (Maxwell Air Force Base / Montgomery)
Last Reported Time: 2055
Last Reported Position Coordinates:
49 Deg 47 Min N Latitude
56 Deg 16 Min W Longitude
Percent of Flight Complete: 55.32

Flight Number: MC0802
Flight Departed Yesterday
Departure Location: OIZJ (Jask)
Destination Location: BIX2 (Biloxi, Keesler Air Force Base, Navu)
Last Reported Time: 0015
Flight Arrived at: 0015
Reached Terminal at: 0023

Flight Number: MC0803
Departure Time: 1215
Departure Location: BIX2 (Biloxi, Keesler Air Force Base, Navu)
Destination Location: UHPP (Petropavlovsk-Kamchatskij)
Last Reported Time: 1927
Flight Arrived at: 1927
Reached Terminal at: 1937

Flight Number: MC0903
Departure Time: 1630
Departure Location: FZEA (Mbandaka)
Destination Location: RCDC (Pingtung South Air Force Base)
Last Reported Time: 2045
Last Reported Position Coordinates:

16 Deg 28 Min N Latitude
61 Deg 54 Min E Longitude
Percent of Flight Complete: 45.47

Figure 7. Typical email Message

To pull information from the site, the user logs onto the site and from the switchboard page
shown in figure 4. The user would then choose the Flight Query button. This action will bring
up a web page for the user to formally log into the system for retrieving information (figure 8).
The user will enter first and last name and activate one of three options: Map Flights, Time Line,
or Reset Values. The last option will allow them to re-enter a correct version of their name if it
was misspelled. If the user chooses Time Line, the system will return a page similar to that
shown in figure 9, where each row represents either a flight departed that day or a flight departed
but not landed the day before. This picture allows the user to view the respective time lines of all
flights for that day based on the current time. The mission numbers of each flight are shown
along with the current time, indicated by the green vertical line.
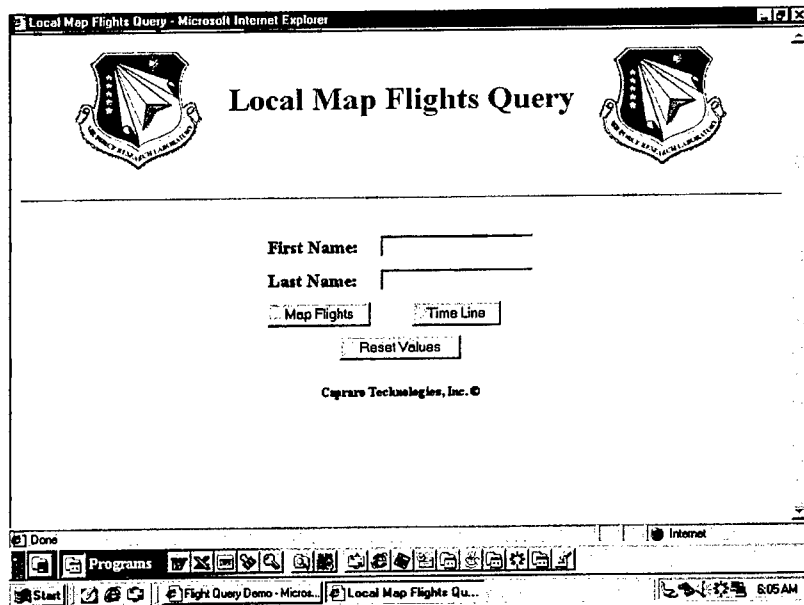
Figure 8 Flight Query Page
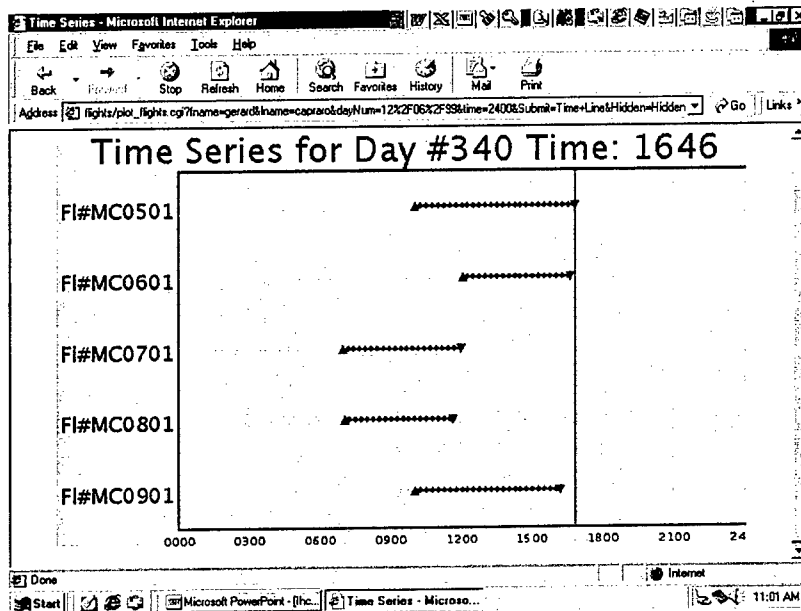
Figure 9 Time Line Page

If the user would like more detail on each flight they can choose the Map Flight button shown in figure 8. This action will provide the user with a world map highlighting each of the flights that were active for that day with their latest position (see figure 10). Each flight is shown using a different color. The user can obtain more information about each flight by clicking on any one of the paths shown on the map. This action will display, for example, the information shown in figure 11. From this page the user can choose one of two buttons to activate. These actions will provide information related to the crew or the passengers as shown in figures 12 and 13 respectively.
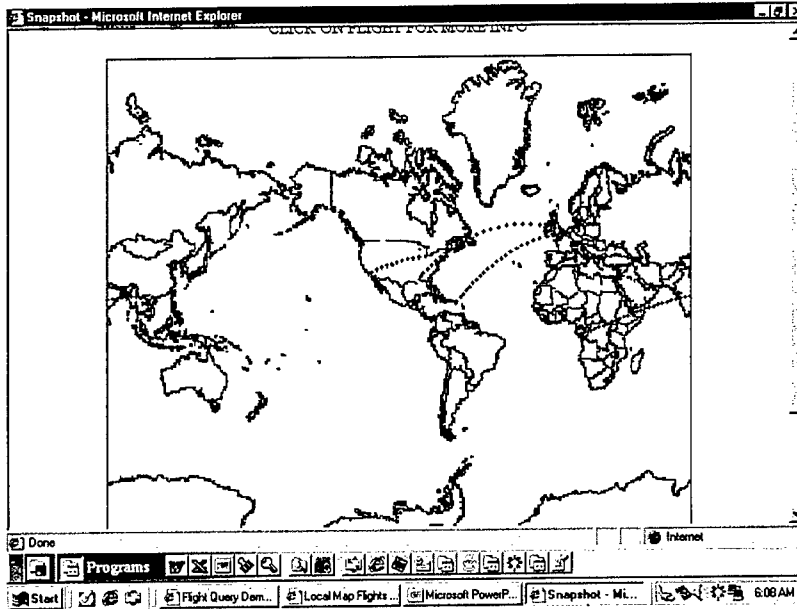
Figure 10 Flight Map Page
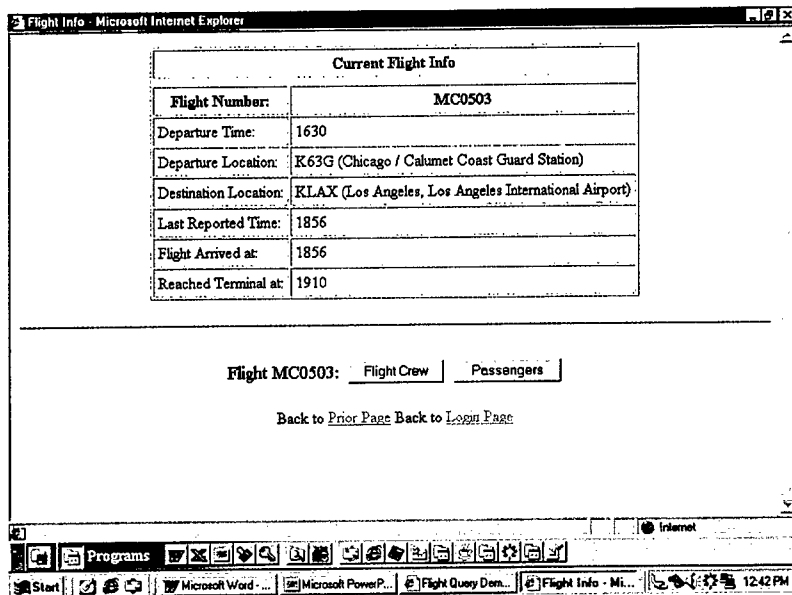
Figure 11 Flight Information

Figure 12 Crew Information

Figure 13 Passenger Information

If the user enters the system with a device that has a browser front end but is limited in screen capability (e.g. Windows CE handheld device) or would rather have the information provided in text form, then they can enter their first and last name as "h" and "hcd" respectively (see figure 8). Then if they activate either the Time Line or Map Flights the user will get a response similar to what is shown in figure 7.

In addition to pushing information using email and pulling information using web enabled devices we can also provide a user the capability to pull information using their email. This works well whether the user is pulling the information with a Palm device or a PC. To activate this capability the demonstrator clicks on the Mail Monitor button shown in figure 4. Once activated a user can send an email to a special mailbox on our site with one of three entries for the subject of the email. These are Flight mcxxxx, or Passengers mcxxxx, or Crew mcxxxx. The user substitutes the last four digits of the flight number of interest (xxxx). Once our mail server receives the message, it is retrieved by a Java mail monitoring program. Another Java program saves the sender's email address, parses the subject line, queries the database and sends the query results back to the requester. The following three figures are representative responses to requests for Flight mc0503, Passengers mc0803, and Crew mc0803.

Flight Number: MC0503
Departure Time: 1630
Departure Location: K63G (Chicago / Calumet Coast Guard Station)
Destination Location: KLAX (Los Angeles, Los Angeles International Airport)
Last Reported Time: 1856
Flight Arrived at: 1856
Reached Terminal at: 1910

Figure 14 email Flight Request

Passengers on Flight: mc0803

0. Private Jennifer Johnson
Phoenix Navel Station
Phoenix, AZ

1. Private Roger Septoff
USMA at West Point
New York, NY

2. Officer Raymond Steinberg
US Coast Gurad Station
Southfield, MI

3. Officer Joseph Riscili
USAF Flying Academy
Amherst, NY

4. Officer David Hanson
Canada Air Station
Mississauga, Ontario

5. Officer Gordon Lamb
Royal Marine Station
Tyne & Wear, Newcastle

6. Private Jim Campbell
McDonald's Air Force Station
Glasgow, null

Figure 15 email Passengers Request

Crew For Flight: mc0803

Navigator: Lieutenant John Gross
Baltimore Navel Yard
Baltimore, MD

LoadMaster: Sergeant Gary Johnson
Kauai Air Station
Kauai, HI

CoPilot: Captain Leonard Croth
Canada Air Station
Mississauga, Ontario

Pilot: Captain Renee Capraro
Fort Hill
Camp Hill, PA

Figure 16 email Crew Request

7.0 Summary, Conclusions and Future Work

This report has provided a description of our efforts to demonstrate the feasibility of integrating AI technology with web technology to bring very large data and knowledge bases to a hand-held computing device (HCD) in an efficient manner. In pursuit of this objective we have performed a literature review of relevant technologies and HCDs. This is presented in Appendix A. We have also described our software architecture and demonstrated its capability using a USAF problem domain. Our software architecture is currently built on top of a relational DBMS and consists of a personal assistant, a hardware assistant and a SQL assistant. The Air Mobility Command of the USAF provided our problem domain. They are interested in capturing information related to all their daily flights. We simulated the basic messages (departure, position, and arrival) for any particular flight given its departure and destination ICAO locations. From this information we demonstrated how a user could set their preferences of how they wish to receive messages regarding departures and arrivals either periodically or based upon a particular event occurring. We also presented how the user could acquire information through pull technology by using a web browser on any email enabled device (e.g. a palm pilot or a workstation). We performed this demonstration numerous times over the web and have provided secure access on our web site for USAF demonstrations.

We have shown that the technology is available today to bring very large data and knowledge bases to a HCD. Since this effort started the JBI information management system has progressed to the point that the USAF has efforts both in-house and through numerous contracts with industry and universities throughout the country. Most of these efforts are concerned about the technologies required to build the JBI infrastructure. Our effort is concerned with serving the clients and their interface to a JBI node. We believe that the approach presented here should be further investigated and continued such that the architecture we build is domain and platform independent. The software architecture described above is only a prototype and is built upon a relational DBMS using SQL and Java. We believe that this approach is sound and should be continued.

There are numerous efforts being pursued by RL/IF and DARPA where ontology modeling technologies are being developed that will allow for the integration of heterogeneous intelligent processes to communicate with each other. The technology they are pursuing will eventually allow for inferencing across ontologies over an Intranet, a JBI architecture, or Internet. It is recommended that this modeling approach be investigated to implement the varied assistants within the prototype architecture described here in. It would be a significant advantage for the personal assistants.

8.0 Acknowledgements

understanding AMC's problem domain. Last on our list we owe a debt of gratitude to the employees of Capraro Technologies, Inc. that provided their talents and creativity in building the prototype software described in this document. These people are: Ms. Sarah Schiavone, Mr. Mark D'Agostino, Mr. Christopher Capraro, Mr. Gerald Berdan, Mr. Anthony Macera, Mr. Brice DeWire, and Mr. Ash Patel. Thank you all.

9.0 References

1. Capraro, G.T., "A Data Management System Problem Specification Model," RADC—TR—73-193, June 1973.

2. Capraro, G.T., Berra, P. B., "A Data Base Management Problem Specification Model, "AFIPS Conference Proceedings," Vol. 43, 1974, pp. 53—56.

3. Capraro, G.T., "A Data Base Management Modeling Technique and Special Function Hardware Architecture," Unpublished Doctoral Dissertation, Syracuse University, February 1978.

4. Capraro, G.T., Berra, P.B., "A Data Base Management Modeling Technique and Special Function Hardware Architecture," RADC—TR—79—14, January 1979.

* 5. Lazzara, L. V., Marcinkowski, J. M., Capraro, G.T., and White, R. C., "Knowledge Based and Database System Integration: A Base Line for a Design Methodology," RADC-TR-86-132, September 1986.

6. Lazzara, L.V., Marcinkowski, J., Capraro, G.T., Liuzzi, R., and White, R., "A Framework for Extending Information Systems With Knowledge Based Processing Capabilities," Proceedings of the Computer Software and Applications Conference, October 1986.

*7. Lazzara, A. V., Marcinkowski, T. M., Tepfenhart, W., Capraro, G.T., White, R.C. and White, T. M. "Data Architecture Concepts for Knowledge Based Systems, KBS and DBMS Integration, " RADC—TR—88—245, October 1988.

* 8. Lazzara, A. V., Marcinkowski, J. M., Decker, J. M., White, R.C., White, T. M., Capraro, G.T., Darras, D., and Cheung, C., "KBS—DBMS Integration Analysis and Design Alternatives," RADC—TR—87—91.

9. Capraro, G.T., Cheung, C., and Liuzzi, R. A., "Large Knowledge Based Systems: An Efficient Approach," Proceedings of the Annual AI Systems in Government Conference, March 1989.

10. Capraro, G.T., Siarkiewicz, K., "Computational Electromagnetics' Future Database Architecture," 13th Annual Review of Progress in Applied Computational Electromagnetics, March 1997.

*Although this report references the above limited technical reports, no limited information has been extracted.

11. Capraro, G.T., "Integrated Computational Environment (ICE)," RL-TR-97-70, August 1997.

12. Kwasowsky, B., Capraro, G. T., Berdan, G. B., Capraro, C. T., "Remote Data Entry and Retrieval for Law Enforcement," First Annual Symposium on Enabling Technologies for Law Enforcement and Security, November 1996.

13. Capraro, G. T., "Hand-Held Computing Devices and Large Knowledge Bases," AFRL-IF-RS-TR-1998-205, November 1998.

14. United States Air Force Scientific Advisory Board, "Report on Informatin Management to Support the Warrior," SAB-TR-98-02, December 1998

15.United States Air Force Scientific Advisory Board, "Report on Building the Joint Battlespace Infosphere," SAB-TR-99-02, November 1999

Appendix A

A TECHNOLOGY SURVEY FOR HAND-HELD COMPUTING DEVICES

BY

GERALD B. BERDAN
GERARD T. CAPRARO

TABLE OF CONTENTS

Introduction

This report is a survey of the state of current technology in the following four areas: intelligent agents; networking and Internet technologies; handheld computing devices; large knowledge bases and databases.

Intelligent agents are the subject of a great deal of current research. Intelligent agent technology is a continuation of artificial intelligence. There is a lack of consensus as to the definition of the term itself, so we present a listing of various attributes that have been ascribed to intelligent agents. From this list, we will use the parts of an intelligent agent that meet the needs of this project. We look at the current implementations of this technology as well, reporting on projects where the abundant research has actually led to products.

The Internet has captured the imagination of the world, and combined with networking technologies, has presented tremendous opportunities to increase utilization and efficiency in many areas of computer technology. Networking is now faced with problems of interoperability and bandwidth limitations, and we look at some of the areas where these problems are being addressed. There is also an opportunity to add intelligence to the network, creating a system where data is gathered intelligently from any source that has relevance, and the system returns all the data and knowledge the user needs.

If computer hardware catalogs are any indication of popularity, handheld computing devices are beginning to sell in significant numbers. These devices are mostly used as electronic organizers for an individual, but they have a tremendous potential to move real computing power to a device small enough to fit in a pocket. When these devices can be linked to a network intelligently, the technology can realize the power of network computing on the individual device.

Data has been electronically gathered and stored for decades, and has grown to immense proportions. Data is gathered at rates that have superseded our ability to use these data in their entirety. Research and products are being considered and produced that can tap this vast store of data in the multitude of formats and locations where these data exist. When accessed, the data has to be formatted for effective use in any system where the use of the data would enable or enhance the current system.

The research in these areas forms the basis for the next task of this phase of the project on the part of Capraro Technologies, Inc. that combines these technologies. This project will utilize a hand held device to connect to a network, access multiple databases across a network using intelligent processing, and return appropriate and scaled data to the hand held device. By demonstrating this capability, we intend to show the ability of the network to use intelligent processing to scale data transmission and retrieval based on the abilities of the connected device to send, receive and process these data efficiently.

26

INTELLIGENT AGENTS

Definition

The history of scientific research finds that at various times, specific areas of research become fashionable within a discipline. These topics become a magnet for researchers, as there are numerous undefined and unexplored areas for substantial and interesting research that can lead to significant breakthroughs. The field of intelligent agents is in this state now; it is a topic that is producing abundant academic and commercial research.

Intelligent agents, although they are generating a lot of current interest, are not new, and, in fact, they date back to some of the earliest computer science research.

> "The idea of an agent originated with John McCarthy in the mid-1950's, and the term was coined by Oliver G. Selfridge a few years later, when they were both at the Massachusetts Institute of Technology. They had in view a system that, when given a goal, could carry out the details of the appropriate computer operations and could ask for and receive advice, offered in human terms, when it was stuck. An agent would be a 'soft robot' living and doing its business within the computer's world." (Kay 1984).

This would be the optimal place in this report to define an intelligent agent, but we are faced with a problem: no consensus definition of an intelligent agent exists. "There are almost as many definitions of 'agent' as there are researchers" (Parunak, 1998).

Rather than a simple definition, it may be better to list concepts that fall under the "intelligent agent" umbrella. Jeffrey Bradshaw has performed an extensive survey of the academic research in the area and has listed some of the key concepts that researchers have utilized in looking at intelligent agents. In a general way, an intelligent agent can, and perhaps should contain the following:

- **Reactivity:** the ability to selectively sense and act.
- **Autonomy:** goal-directedness, proactive and self-starting behavior.
- **Collaborative behavior:** can work in concert with other agents to achieve a common goal.
- **"Knowledge-level" communication ability:** the ability to communicate with persons and other agents with language more resembling human-like "speech acts" than typical symbol-level program-to-program protocols.
- **Inferential capability:** can act on abstract task specification using prior knowledge of general goals and preferred methods to achieve flexibility; goes beyond the information given, and may have explicit models of self, user, situation, and/or other agents.
- **Temporal continuity:** persistence of identity and state over long periods of time.
- **Personality:** the capability of manifesting the attributes of a "believable" character such as emotion.

- **Adaptivity**: being able to learn and improve with experience.
- **Mobility**: being able to migrate in a self-directed way from one host platform to another. (Bradshaw, 1997)

All researchers do not agree that all of these concepts necessarily belong in the definition, but a majority of this set of attributes is found in the many definitions of intelligent agent software. But by allowing such a dynamic definition, just about any software can be described as an intelligent agent. By judicious selection from this list of attributes, a developer can produce an ad hoc definition that fits the needs of the application. By allowing this kind of license, it makes sense to narrow the definition to a few core attributes, and leave the others as optional.

Since there is no consensus, there is a wide variety of software code that has been called an "agent" by developers. These programs can:
- be scheduled in advance to perform tasks on a remote machine
- accomplish low-level computing tasks while being instructed in a higher-level programming language or script
- abstract out or encapsulate the details of differences between information sources or computing services
- implement a primitive or aggregate "cognitive function"
- manifest characteristics of distributed intelligence
- serve a mediating role among people and programs
- perform the role of an "intelligent assistant"
- migrate in a self-directed way from computer to computer
- present themselves to users as believable characters
- speak an agent communication language
- be viewed by users as manifesting intentionality and other aspects of "mental state" (Bradshaw, 1997)

Researchers are faced here with an area that is producing significant amounts of research and development, but yet lacks a consensus definition. Each concept has its proponents and detractors, and until a broad consensus emerges from the melee, researchers will have to work within a broad set of contentious issues.

The next following points are attempts at defining intelligent agents:

- "An agent is a computer system situated in some environment, and that is capable of *autonomous action* in this environment in order to meet its design objectives...The system should be able to act without the direct intervention of humans (or other agents), and should have control over its own actions and internal state" (Wooldridge and Jennings, 1998)
- "An intelligent agent is a computer system that is capable of *flexible* autonomous action in order to meet its design objectives. By *flexible*, we mean that the system must be:

1. *responsive*: agents should perceive their environment (which may be the physical world, a user, a collection of agents, the Internet, etc.) and respond in a timely fashion to changes that occur in it,
2. *proactive*: agents should not simply act in response to their environment, they should be able to exhibit opportunistic, goal-directed behavior and take the initiative where appropriate, and
3. *social*: agents should be able to interact, when they deem appropriate, with other artificial agents and humans in order to complete their own problem solving and to help others with their activities." (Wooldridge and Jennings, 1998)

- "We do not think of agents as invoking methods (actions) on agents – rather, we tend to think of them *requesting* actions to be performed." (Wooldridge and Jennings, 1998)
- The emphasis of software agents "has subtly shifted from deliberation to doing; from reasoning to remote action." (Bradshaw, 1997)
- The best agents, then, would not only need to exercise a particular form of expertise, but also take into account the peculiarities of the user and situation. (Bradshaw, 1997)
- We expect an agent that inhabits an environment with other agents and processes to be able to **communicate** and **cooperate** with them, and perhaps move from place to place in doing so. (Bradshaw, 1997)
- A more specific definition of "software agent" that many agent researchers might find acceptable is: a software entity which functions **continuously** and **autonomously** in a particular environment, often inhabited by other agents and processes (Shoham, 1997).

One conclusion that can be reached from this mixture of definitions and components is that an "intelligent agent" is always based on a personal perspective that will not always agree with either the developer or the intended user. It is also important that the fundamental research be continued and that eventually a consensus be reached as to what this technology is, and what direction it is going.

> "If agent technology is to achieve its potential, then these pragmatic aspects of agent system development must be studied and understood -just as they have been for object-oriented programming. There is a very real danger that if no attempt is made to do this, then agent technology will fail to live up to the claims currently being made of it. The result will be a backlash similar to that experienced against expert systems, logic programming, and all the other good ideas that, it was promised, would fundamentally change computing." (Wooldridge and Jennings, 1998)

It is ultimately very important to recognize that "intelligent agents" are not a new and independent area of computer science, but are a branch of artificial intelligence. The research arises from AI theory, and combines many of these attributes with those of object-oriented technology.

Current Implementations

Research in intelligent agents has generated quite a few implementations in software. There are two types of implementations: agent projects and agent technology. Agent projects are stand-alone programs that function as an agent for the user, whether individual or corporate. Agent technology is modules that fit into other software projects to perform agent activities for these programs.

In all of the areas where agent technology is useful, there seems to be at least 1 common attribute: a dynamic situation that requires reactive and proactive software. These programs infer high level goals from user actions and requests, allowing human imprecision, but redefining these high level goals within the dynamic structure.

Björn Hermans has defined 8 areas where agent technology is either implemented or in development. They are:

1. Systems and Network Management. " In the face of rising complexity" administrators must utilize an increasing amount of software in order to manage huge networks. The era of the autocratic and independent network administrator is ending.
2. Mobile Access/Management. Users are no longer chained to a workstation: they demand mobility. "Intelligent agents...reside in the network rather than on the users' personal computers, can address these [computational] needs by persistently carrying out user requests despite network disturbances. In addition, agents can process data at its source and ship only compressed answers to the user, rather than overwhelming the network with large amounts of unprocessed data."
3. Mail and Messaging. Agents can facilitate e-mail by rules implementation. These rules are dynamic and user-driven.
4. Information Access and Management. Internet and enterprise-wide information retrieval will require substantially more intelligence than current search engines employ. A massive amount of data in response to simple queries is almost useless, and needs to be intelligently filtered.
5. Collaboration. " Not only do users in this area need an infrastructure that will allow robust, scaleable sharing of data and computing resources, they also need other functions to help them actually build and manage collaborative teams of people, and manage their work products."
6. Workflow and Administrative Management. Another dynamic area, where "intelligent agents can be used to ascertain, then automate user wishes or business processes."
7. Electronic Commerce. "Intelligent agents can assist in electronic commerce in a number of ways. Agents can "go shopping" for a user, taking specifications and returning with recommendations of purchases which meet those specifications. They can act as "salespeople" for sellers by providing product or service sales advice, and they can help troubleshoot customer problems." Electronic commerce is not limited to commercial business, but its activities can be applied to any type of large-scale distribution of materials, including government agencies.

8. Adaptive User Interfaces. "Intelligent agent technology allows systems to monitor the user's actions, develop models of user abilities, and automatically help out when problems arise."

Many of the commercial intelligent agents currently available are Internet-based programs that are designed to retrieve information and collect data, such as SandPoint's Hoover. Information retrieval software agents can view data from around the Internet based on an individual's "profile," a dynamic set of user preferences, wants, and needs. This data can be packaged for view, prioritized, stored, and otherwise processed based on the profiles. This type of agent can retrieve data and information from a set of defined sites, use rules to find new sites that may contain pertinent information, "learn" from its experiences to change the profile, keep track of time-sensitive data for updates, and several other "agent" functions. The profile is not necessarily explicitly entered by the user, but the software agent monitors the users reactions to results and can change priorities based on this experience.

Another type of agent program extracts information about individuals without their knowledge. This type of agent collects data on the people who visit web sites:
- Who they are,
- Where they came from,
- Where they go to,
- What kind of software they are using,
- What kind of hardware,
- How many times have they visited,
- How long they look at various pages within the web site,
- How they navigate within the web site, and several other data.
A use for this data may be to create specialized mailing lists for advertisers, or to see the effectiveness of advertising on a web page.

Another specific area in which agent technology is being implemented is in pattern recognition software. Speech and handwriting recognition involve a dynamic learning process that needs to be able to correct its inevitable errors for an individual subject, and be able to accommodate new users.

There are many implementations of agent software and agent technology in commercial, academic and government areas. We have listed a few interesting examples.

IQ software has produced a suite of intelligent agent programs to be used by enterprises for decision support. This suite of agent-based programs performs a variety of actions to meet information needs within an enterprise. The programs include these capabilities, among others:
- access multiple databases (in many formats),
- access and mine data from internet and intranet pages,
- perform many different types of queries across platforms,

- run time-sensitive queries or persistent queries that only return data if certain conditions are met at certain times,
- contain multi-dimensional data storage and retrieval
- data dimension definition on the fly
- scheduling
- on-demand reporting
- "what-if" scenarios across the enterprise

An agent project that has grown from academic research is The Stanford Digital Libraries Project. They have developed a program called InfoBus that is used to create a virtual library that theoretically may have no limits. It is based on what the developers have called "Distributed object technology," which is pulled together and presented through a unified interface using intelligent agent technology.

> "In an ideal world, clients and service providers that are part of a digital library would be created independently, on the basis of implementation choices the respective consumers and providers deemed appropriate. Then everyone would plug their components into a virtual software bus that would take care of all the protocol-level interoperability issues. Within this information bus (which we call InfoBus), library services would transparently translate formats, broker services, and support financial transactions. If all services conformed to one standard, the developers of digital libraries could easily realize this vision. Unfortunately, protocol convergence has not occurred, even in the long-standing area of information retrieval. An overly simple solution would call for cross-translations among all standards. This would be a formidable effort. Distributed object technology may help achieve the long-term goal of an InfoBus without requiring all participants to agree on a single standard mode of interaction. (Paepcke 1998)"

This project demonstrates the independence and modularity that are integral attributes of object-oriented technology and intelligent agent technology

There are also some very simple agents. for example: Apple's Apple Data Detectors (Nardi). It searches text, utilizes a grammar to find a URL anywhere in this text, then connects the browser to that web site.

Other projects listed in the Agent News WebLetter 3.02
- CyberLife Technology is working with the UK's Ministry of Defense (MOD) research organization "to build a simulated military aircraft controlled by a software agent. CyberLife will be using real flight model data from the MOD to simulate an aircraft akin to the Eurofighter. "This intelligent plane, however, requires no human intervention and will be capable of sustaining flight, pursuing enemy vehicles, evading attack and making reasoned decisions in order to complete its mission requirements."
- "The Do-I-Care Agent (DICA) developed at UC [University of California at] Irvine addresses the resource re-discovery problem on the Web. Once you've found an interesting

site, how do you know when new and interesting material has been added? For example, you may want to know about a new and interesting paper by a colleague. Or you may want to know about any airplane ticket sales to Australia. You don't want to know about minor changes - you want to know when cheap fares to Australia are available. DICA solves this problem by periodically visiting the site and only informing you when something interesting has occurred. You provide DICA with feedback on its interestingness judgments and thus train it to recognize the changes of interest to you. "

- "Intelligent Reasoning Systems (IRS) is a small R&D and consulting company based in southern California which specializes in Intelligent Software Agents and Distributed Artificial Intelligence. IRS offers two systems for downloading -- Jam! and UMPRS. Jam! is a Java-based intelligent agent architecture that grew out of academic research and extended during the last five years of use, development, and application. Jam combines the best aspects of several leading-edge intelligent agent frameworks, including the Procedural Reasoning System (PRS) and SRI's ACT plan interlingua. UMPRS is a C++ implementation of a PRS-like planning engine developed at the University of Michigan." (Agent News WebLetter 3.02)

NETWORKING AND INTERNET TECHNOLOGIES

The World as Distributed Information System

The mainframe world of the 1970s has been superseded by a new computational model consisting of vast numbers of independent nodes of heterogeneous systems. These systems are linked in a variety of ways. The average network now usually consists of many processors, each working independently, multiple storage sites, sharing of applications, held together in various degrees of cohesiveness and conformity.

Beyond the local network (LAN), there also exist relations between these networks as wide area networks (WANs), and ultimately these WANs, LANs and individuals form the loose affiliation called the Internet. The world of information is now a jumble of processors and storage, growing almost uncontrollably, a leaderless, almost anarchic affiliation. The problems of this great network are problems of scale. The computational models that exist today are inadequate for vast numbers of autonomous hosts. Today's software is unable to scale to these levels of network complexity.

Along with this uncontrolled growth is a growing need by users to access data and information, and to access them efficiently and with minimal latency. The fact that some information is theoretically available to a user is just an irony if the user cannot find it. The latest research looks at how to make this information available, to overcome the inherent problems of a wide distribution of sites and information, across the many millions of processors and drives that comprise this network.

The problems of the Internet are not just problems of speed and throughput. There are problems with data consistency. When there are multiple storage sites for redundant data, how can a system guarantee that queries sent will return consistent results? There are also problems with data accessibility. How does a user (either human or computer) know that relevant information exists that may yield a better, more accurate query result? These problems are not limited to the Internet, but they are problems with networking itself. There is less and less of a difference between the Internet and the network, as networks scale upwards.

Given the problems of availability and consistency, there is a trade-off in that to increase one of these attributes, means a subsequent decrease in the other. Increase the availability of data, and the consistency of query results decreases.

Small networks are not faced with these problems. Bandwidth issues are comparatively easy to solve, the data can be warehoused effectively, and consistency is the norm. The Internet is not simply a matter of stringing wires, but as it grows, it presents significant scaling issues. And these problems are not limited to the Internet, but as organizations pull together their many

disparate data sites into ever larger WANs, then these organizations face their own versions of these problems.

The research in this area is at both macro and micro levels. There are some projects that encompass the entire computing world as a topic. One such project is the Legion Project.
> "Inevitably, users will operate in a wide-area environment transparently consisting of workstations, PCs, graphics-rendering engines, supercomputers, and nontraditional computing devices, such as televisions. The relative physical locations of users and their resources is increasingly irrelevant.(Grimshaw, 1997)"

The Legion Project is designed to provide a "solid, integrated conceptual foundation" towards viewing these vast networks as a "Worldwide Virtual Computer (Grimshaw, 1997)." Begun at the University of Virginia in 1993, this project's goal is "a system consisting of millions of hosts and billions of objects coexisting in a loose confederation united through high-speed links (Grimshaw, 1997)." The technology for this project already exists:
- "Parallel compilers that support execution on distributed memory machines.
- Distributed systems software that manages complex distributed environments.
- General acceptance of the object-oriented paradigm because of its encapsulation and reuse properties.
- Cryptography and cryptographic protocols. (Grimshaw, 1997)"

This system is not limited to traditional definitions of networks. These autonomous hosts can range from the largest mainframe system to hand-held devices, and the objects on the network can be any form of device linked to a computer, including cell phones and digital cameras, among others. This system views the entire world as a single computational resource.

The Legion Project is by no means an isolated attempt at overall Internet integration. Other large-scale projects include Nexus, Castle, NOW, and Globe, each approaching the fundamental problems from different angles. Corba is another example of a standard model that may be useful in overall integration efforts, and Java promises to bring a new level of interactivity. Until science defines the system, scalability is the most important impediment to worldwide integration.

Problems:
Along with the overall scalability issue, there are many other individual problem areas now being researched. There are many areas where there are significant problems and significant research. These areas include: algorithms, design, management, measurement, performance, reliability, security, and standardization. We have included a discussion of some of these aspects.

Interoperability
If there is to be a "virtual computer", whether worldwide or across an enterprise, then there should be interoperability of those components. Issues of interoperability exist in every aspect of

networking, including both hardware and software. In order to try to come to some reasonable accommodation across various commercial, government and academic organizations, the University of New Hampshire's InterOperability Lab (IOL) exists to verify and certify such interoperability.

The IOL is also the center for various consortiums dedicated to publishing and promoting open standards in various areas of computer and network hardware and software. There are consortiums for: 1394, ADSL, ATM, Fast Ethernet, FDDI, Fibre Channel, Gigabit Ethernet, IP/Routing, Network Management, Token Ring, VLAN, and Wireless.

Interoperability increases the usefulness of COTS and GOTS, allowing what would have been proprietary software to move beyond its original market. Interoperability actually creates markets for these products by assuring that a collection of hardware and software components actually work as a system. Of course interoperability is a goal, and not a reality in the current market. But as the object paradigm continues its move to the forefront of technology, interoperability allows the object model to implement its modular nature and reusability.

Bandwidth

Lack of physical bandwidth is really not among the most compelling problems in networking. Given the appropriate financial resources, the cabling can be provided. Cabling is available almost everywhere, and there is actually more bandwidth available in the overall system than can be used. Renting this excess bandwidth is the next wave, as privately wired WANs share their wires with other WANs.

This is not to say that there are no problems with bandwidth capacity. There are certainly some significant issues with remote devices, such as handheld computers and cellular phones. These are problems with some of the objects on the web, but not especially with the nodes. There are also specific areas where a lack of bandwidth severely limits the functionality of a network. But these technology limitations are at the periphery of networking issues. The most important problems with bandwidth are actually in the software, not the hardware.

Bandwidth Utilization

We cannot just throw additional bandwidth at a problem and expect this additional bandwidth to improve throughput. It is only in certain circumstances that this approach will work. If the real goal is to increase throughput of a network, then the solutions usually lie in software, and not hardware. Efficiency is the issue; its limitation is not solved by merely adding cables to the system.

A significant issue in bandwidth utilization is the problem of congestion. There are many false assumptions that many make in dealing with the congestion issues. Jain has shown that the following statements are false:

"1. Congestion is caused by a shortage of buffer space. The problem will be solved when the cost for memory becomes cheap enough to allow infinitely large memories.
2. Congestion is caused by slow links. The problem will be solved when high- speed links become available.
3. Congestion is caused by slow processors. The problem will be solved when processor speed is improved.
4. If not one, then all of the above developments will eliminate the congestion problem. These old myths are based on the belief that as resources become less expensive, the problem of congestion will automatically be solved...Increasing memory sizes, processor speeds, and link bandwidths has actually aggravated the congestion problem. Proper inclusion of congestion management and avoidance mechanisms in protocol design is more important today than ever before.(Jain 1992) "

Although this basic research is somewhat dated, the ramifications of the research are only now being felt. You cannot eliminate congestion by throwing hardware at the problem, the problems of software, protocols and several other areas must be addressed as well.

There are two approaches to the problem of congestion: Congestion avoidance and congestion management. There are significant problems of congestion when there are heterogeneous network interconnections. These problems were far easier to solve in small networks, but contemporary reality now assumes a great deal of heterogeneity.

Congestion avoidance is a "prevention mechanism", and congestion control is a "recovery mechanism" (Jain & Ramakrishnan, 1998). Congestion avoidance prevents a congested state, and congestion control manages a congested state and allows the system to recover from it. Congestion avoidance is difficult where arrival rates, service requirements and priorities are difficult to anticipate such as in the large network. Congestion control is now the focus of significant research, because many of the avoidance schemes slow throughput of the network. Unless one is able to assume a congestion-free system most of the time, and develop methods for recovery, the always congestion-free system must be so conservative as to degrade performance.

Measurement

A significant research area is that of network measurement. There are systems that measure every aspect of networking. Measurement is not just a matter of throughput. Throughput gives some number that is useful in heterogeneous situations. But we should measure not just how much data gets through, but what those data are, who got it, how critical was it and did it get through in time. There are actually some data that can come too quickly, as occurs with timing issues in parallel processing across networks.

Other areas of measurement include:
• Rate calculation across networks is, perhaps, one of the most significant areas in measurement research. Congestion is caused by traffic bursts, and we need to be able to calculate the heterogeneous rates.

- Anticipation allows congestion avoidance by pre-determining needs based on various algorithms, probabilities and use patterns.
- Bottleneck speed is a critical measurement. It can be critical to find the bottleneck area. Cross network bottleneck measurement is an area of opportunity for research.

Measurement tools, such as Bprobe and Cprobe perform some of these measurements. Developed at Boston University, these tools can be used in these measurements. Bprobe measures the level of available bandwidth, and Cprobe measures the level of congestion. Their purpose is to allow a host the choice of a server.

Unfortunately, these tools can carry so much overhead as to cause more congestion than they actually save. This is a common shortcoming of all measurement tools, in that the better the measurement, the more CPU and bandwidth the system uses, and congestion becomes more likely.

High Bandwidth Solutions

It has been a long time since Ethernet was born, and there are now well over 100 existing network technologies. Each of these is meant to be an improvement on the others, and they have, in general, increased networking capabilities.

There has always been a quality vs quantity dilemma in networking. Speed is only valuable when the data received is usable. It is important to consider Quality of Service (QoS) requirements in all of these areas.

ATM, or asynchronous transfer mode, is a technique to transport data that improves on the efficiency of earlier methods. Its development reflects the need for more efficient networking protocols and the need to increase the capacity of networks beyond purely hardware solutions. First developed in the early 1980s, it works by transferring small fixed-size cells (53 bytes) over virtual connections (VC). This allows for small, predictable transmissions, and with the small, fixed sizes, the transmissions can be allocated more easily. The ultimate purpose of this system is to fully utilize bandwidth. Increasing bandwidth is physically simple. Utilizing bandwidth efficiently is difficult.

ATM utilizes cells with extremely small headers (comparatively) of 5 bytes for addressing and connection data. Because it is small, there is substantially less overhead in the transportation, that is, fewer bits are actually sent. And the header also has an area within these 5 bytes for a QoS indicator. This indicator can be used to set priorities at the cell level. ATM creates virtual channels that carry cells of the same QoS level within them. If transmissions occur with 2 different QoS levels, then 2 different virtual channels are created. This actually simplifies the switching through the network.

Advantages of ATM are that the system is flexible and configurable, and it is designed to support different kinds of network traffic with different requirements and priorities. Because it supports 3

different classes of cell exchange it is extremely flexible, and can handle different types of data in different ways.

> "To promote efficiency for applications of all types, ATM is partitioned into several different cell-relay service classes. The functions and key characteristics of these service classes are described as follows:
> CBR -- Continuous Bit Rate service provides constant bit rate capacity at a bit rate specified by the user for applications such as voice, video, and circuit emulation.
> VBR --Variable Bit Rate service delivers traffic of varying bit rate, up to a defined bit rate and provides guarantee of delivery. Within VBR are subservices called real-time VBR (RT-VBR) and non-real time VBR (NRT-VBR). These subservices primarily address different levels of delay variation control for certain applications.
> ABR -- Available Bit Rate service supports unpredictable and bursty flow of cells from a transmitter to one or more receivers. This service can support elastic type LAN applications that are relatively insensitive to variations in delay and have a low tolerance for cell loss. For applications such as LAN Emulation that can tolerate delay, but no cell loss, ABR can be a highly valuable service offering. (Quantum Flow Control Corp, 1998)"

This flexibility and configurability is combined with the efficiency of small, fixed-size cell switching, is significant in promoting ATM to the networking world.

ATM is not perfect. It is just another step in increasing network efficiency. Some of its problems are detailed in a marketing report from Sage Marketing. It lists the high cost of hardware, both NICs and routers, the lack of software management tools, and the cost to retrain personnel to use it (Sage Research, Inc, 1998).

There are several commercial hardware/software products available on the market that are geared towards promoting the speed of networks to the gigabit per second level. These products, such as Myrinet, are actually just trying to get networks to keep pace with the processing speed of its nodes.

There are still other new networking standards being promoted: these are standards, not products, and are being implemented by various organizations. Gigabit Ethernet is the latest version of Ethernet. It supports the 1 gigabit per second transfer rate, and its first standard was ratified in 1998 by IEEE 802.3 Committee.

Intelligent Networking

Intelligent networking is the ability of network software to employ data from various sensors to increase the efficiency of a network. It utilizes artificial intelligence, and its child, intelligent agents, and employs a rules-based system that increases network efficiency. Users on a network perceive a network as efficient if he/she receives the data/information that the user needs, when he/she needs it. These data should be intelligently gathered from around the "worldwide virtual

computer", and the system should get all the data the user needs, from whatever sources are available.

As we have seen, just pushing data through wires is not network efficiency. We need to incorporate intelligent strategies in data transfer, priorities, security, and timing in order to manage these heterogeneous virtual networks. It is here that the intelligent agent systems can be utilized to increase the efficiency of a network.

An area of research that moves towards intelligent networking is in the field of agent communications languages. Agent communications languages are agents whose purpose is to run autonomously across networks. One such agent communications language is Knowledge Query Manipulation Language (KQML). One may think of these as another instance of cross-platform, cross-network languages, such as HTML (Hypertext Markup Language), Java, VRML (Virtual Reality Modeling Language), DHTML (Dynamic Hypertext Markup Language) and others.

KQML is a language and protocol developed as part of the ARPA Knowledge Sharing Effort. Its purpose is to exchange data and knowledge across platforms in order to facilitate the creation of large knowledge bases. It consists of primitives (called performatives) that express attitudes regarding the content of the exchange and allow agents to communicate such attitudes to other agents and find other agents suitable to process their requests. It coordinates interactions with other agents across these networks.

It is clear that in order to link heterogeneous hosts into a "virtual computer" a great deal more research needs to be completed.

Distributed Processing to Increase Throughput

Another area of research is in distributed processing. The "worldwide virtual computer" would be far short of its goal if it only involved data/knowledge sharing. There are not only many data storage sites, but there are even more processors networked into the system. To truly increase network efficiency, we must increase processor efficiency, taking advantage of unused cycles and parallel processing across the network.

There are many ongoing projects that perform this function. One of them is Condor, a software system developed at the University of Wisconsin. It is important to note that this is software, and another example that hardware is outpacing the ability of software to keep up with it, especially for a huge networking project. Meant to run in a "high throughput computing" environment.

> "Condor is a software system that runs on a cluster of workstations to harness wasted CPU cycles. A Condor pool consists of any number of machines, of possibly different architectures and operating systems, that are connected by a network. To monitor the status of the individual computers in the cluster, certain Condor programs called the Condor "daemons" must run all the time. One daemon is called the "master". Its only job is to make sure that the rest of the Condor daemons are running. If any daemon dies, the

master restarts it. If a daemon continues to die, the master sends mail to a Condor administrator and stops trying to start it. Two other daemons run on every machine in the pool, the "startd" and the "schedd". The schedd keeps track of all the jobs that have been submitted on a given machine. The startd monitors information about the machine that is used to decide if it is available to run a Condor job, such as keyboard and mouse activity, and the load on the CPU. Since Condor only uses idle machines to compute jobs, the startd also notices when a user returns to a machine that is currently running and removes the job (University of Wisconsin, Computer Science Department, 1998)."

HANDHELD COMPUTING DEVICES (HCDs)

Hardware

A Handheld Computing Device (HCD) is a small, portable computing device with a processor, memory, storage ability, user input, visual output, and the ability to run different programs. The latest HCDs are significantly more powerful and faster than any hand-held device that has come before. The latest versions of palm-sized computers feature more processing power and memory than did full desktops of just 10 years ago. And no longer is there a clear distinction between a laptop and an HCD, as there are devices at every level from the smallest palmtop upward.

In Appendix A, we have listed 24 different HCDs along with their attributes. This list is a description of the state of the HCD technology as of early December 1998. It only lists the attributes of a basic system. Because many of these machines can use PCM/CIA cards and serial ports, we can increase the capabilities of any individual machine by adding third party cards. Some of the hardware advances for HCDs are actually technology advances in these peripherals.

Smart Phones

Cellular phones now contain the processing power of an HCD. There are new categories of pagers, cellular phones, digital phones and satellite phones. Communication devices are now able to send, receive, process and store data. And they can actually run user-directed programs. With improved cellular standards and digital communications, data transfer is now possible at a reasonable speed.

The Nokia 9000il is not just a cellular phone, but it opens up to a keyboard and screen, running e-mail, organizer and scheduler software, and even browsing the Internet (although poorly). It is taking advantage of an improved cellular standard, GSM (Groupe Speciale Mobile), a standard that is widely used in Europe, with its better data compression and communication.

The Qualcomm pdQ SmartPhone actually runs the PalmOS HCD operating system. It combines a digital phone technology with the HCD, running any PalmOS software. It includes a touch screen with handwriting recognition, and the ability to synchronize data with your PC.

Battery Life questions

One of the largest problems with HCDs is in the area of battery life. The limitations of the HCDs in Appendix A are quite apparent, with utilization times ranging from 3 hours to over 1,000 hours, but the machines with the most power, connectivity options and screen resolution have battery lives of less than 10 hours. Peripherals, especially PCM/CIA cards consume power at unacceptable rates. With such battery time limitations, many applications for HCDs are difficult.

There is a great amount of research in improving solar power/solar battery technology. If power requirements can be shrunk to the point where small solar cells can sustain HCD performance indefinitely, we may be able to see an explosion of usage similar to that when the small calculator moved to solar power. It is hoped that by using more effective power saving options,

solar chargers, and improved battery technology, the effective "powered-on" time for an HCD can be significantly lengthened. If the power time is lengthened, there can be a significant increase in the number of domains which can utilize HCD technology.

Wearable Computers

Small, portable computer technology is not limited to shirt pocket devices. There is portable computer technology known as the wearable computer. It could be argued that these computers do not fit into the HCD definition, as they are not handheld, but they do merit consideration in technology reviews such as this. The common ground for all these devices is portability, and the wearable computer may be the most portable of all.

The major academic research in this area is being done at the M.I.T. Wearable Computing Project (http://wearables.www.media.mit.edu/projects/wearables/). There are also other research projects at the University of Oregon, Georgia Institute of Technology, The University of Toronto, Canada, and the University of Birmingham, U.K. There is even a research project at the Australian Institute of Marine Science in developing an underwater wearable computer.

The following is M.I.T.'s definition of a wearable computer:
> "To date, personal computers have not lived up to their name. Most machines sit on the desk and interact with their owners for only a small fraction of the day. Smaller and faster notebook computers have made mobility less of an issue, but the same staid user paradigm persists. Wearable computing hopes to shatter this myth of how a computer should be used. A person's computer should be worn, much as eyeglasses or clothing are worn, and interact with the user based on the context of the situation. With heads-up displays, unobtrusive input devices, personal wireless local area networks, and a host of other context sensing and communication tools, the wearable computer can act as an intelligent assistant, whether it be through a Remembrance Agent, augmented reality, or intellectual collectives (Massachusetts Institute of Technology, 1998)."

Wearable computers can actually remember things for the wearer, storing data from sensors as a person moves from place to place, keeping track of time and events. It can be used to retrieve data as well, recognizing people and things, giving real-time warnings or information on the wearer's environment.

This field is growing quite rapidly, and includes a number of international conferences and symposia on the subject. For example, the Second International Symposium on Wearable Computers was held in Pittsburgh, PA. on Oct 19-20, 1998. There are publications devoted to this field as well, such as Wearable Computing Magazine.

Wearables are not a new phenomena. The MIT timeline lists wearable hardware, starting from eyeglasses worn in medieval times, through the early electronic computer developers in the 1960's, through today's research, including DARPA projects, and into commercial development. In 1966, Ed Thorp and Claude Shannon displayed a wearable computer they used to predict

roulette wheel results. DARPA started the "Smart Modules" program in 1994 and 2 years later sponsored the "Wearables in 2005" workshop (Rhodes, 1998).

There are several companies currently producing versions of wearable computers. Here is a short list, from Mediaeater, Inc.:

- Aportis
- Aerocorp Technologies, Inc.
- Advance Systems Ltd
- Franklin Electronic Publishers Inc
- Hang Ware
- Handy Key
- Honeywell Human Computer Interaction
- GeoPerception Inc.
- General Magic
- i-O Display Systems
- Intervision Systems
- Lucent - micro
- Microware
- Motorola - Lexicus Division
- The MicroDisplay Corporation Home Page
- MicroOptical Corp
- NTT Human Interface Labs.
- Orang-Otang
- Philips
- Personal Electronic Devices
- Sentel
- Seattle Sight Systems Inc
- Seiko
- Speech Recognition Technology
- Starfish - True Sync
- Via Inc.
- Virtual Technologies, Inc.
- Virtual Research Systems
- Xybernaut

Software

Windows CE

Windows CE is the most widely used operating system for HCDs. Since its introduction in September, 1996, hardware manufacturers have been building HCDs to run this operating system. To demonstrate market penetration, it is sufficient to note that of the 24 HCDs listed in Appendix A, 16 of them use Windows CE.

Windows CE is not standing still, but is continuing to evolve as the market determines the areas of demand. October, 1998 saw the introduction of a significant version upgrade to this OS.

> "REDMOND, Wash. - Oct. 8, 1998 - - Microsoft Corp. today announced that the third generation of software for Handheld PCs (H/PCs), known as Microsoft® Windows® CE Handheld PC Professional Edition, has been shipped to 12 original equipment manufacturers (OEMs). H/PCs are streamlined, specific-use computing devices designed to be mobile companions that extend Windows operating system-based desktops and notebooks, providing instant access to users' information. Some OEMs will demonstrate new hardware running the H/PC Pro Edition software next week at the Microsoft Professional Developers Conference in Denver, with initial quantities of devices expected to be available in stores later this year, and widespread availability in 1999.
> Formerly code-named "Jupiter," the H/PC Pro Edition software based on the Windows CE 2.11 operating system improves remote access and connectivity to corporate data while providing the familiarity of Windows. In addition, the H/PC Pro Edition software enables new hardware differentiation such as full-size VGA and Super VGA displays and alternative pointing devices such as a mouse (Microsoft, Inc. 1998)."

Some of the significant upgrades to the OS are:
- Java™ support
- Database persistence in memory
- Support for multiple file systems
- Support for multiple protocols

These upgrades, and a commitment on the part of such a large company to this and future upgrades, indicates that there is significant commercial support for these devices, and they will be around for years.

Windows CE is actually developed to work within any system, including imbedded applications. It is built with scalability, so that devices can use pieces of the OS within many different types of devices. An example of this is AutoCE. AutoCE includes a voice technology for drivers to control their audio, navigation, and personal contact systems. It also allows the system to synchronize data with other CE devices, laptops, even desktop computers.

Other HCD Operating Systems

There are other operating systems that run HCDs. One important factor in operating systems is whether there will be applications that the operating system will run. Operating systems that run only custom-built software have proven to fall short in the marketplace, regardless of technical assets.

Two of the more popular HCD operating systems are PalmOS, which runs 3-Com's Palm Pilot family of devices, and the Psion operating system. The devices for these operating systems are very popular, the Palm Pilot being the single largest selling device in the U.S., and the Psion holding the European market. There are also a few other small operating systems, such as Sharp, but these systems are very small, and run only a few applications.

General Magic's Magic Cap, which had been thought of as one of the best HCD operating systems, has almost disappeared from the market. It has been spun off to a new company, DataRover. The DataRover 840 is a HCD that is a data collection/retrieval HCD for vertical markets, running proprietary software. It is an operating system limited for use in HCDs within tight vertical markets in that it has very little third party software support.

The next generation of HCD operating systems may be Java™ based. There is an interesting project coming out of Oracle that may supersede the need for much of an operating system on these devices. This project is described later in this paper.

The following is a list of add-on software & hardware suppliers supporting Windows CE Handheld PC Professional Edition:

Advanced Systems (ASL)
Advanced Recognition Technologies Inc. (ART)
ARM Inc.
AvantGo Inc.
BSQUARE Corp.
Communication Intelligence Corp. (CIC)
Citrix Systems Inc.
C-Labs
Cloudscape Inc.
Eclipse International Inc.
EnCompass Globalization Inc.
Futuresoft Inc.
Hitachi Semiconductor (America) Inc.
Infowave Wireless Messaging Inc.
Inso Corp.
InstallShield Software Corp.
Intel Corp.
Iomega Corp.
Integrated Technology Express Inc. (ITE)

Joey Technologies Inc.
JP Systems
MVA Software Inc.
NEC Electronics
NS BASIC Corp.
Object Design
Odyssey Software Inc.
On The Go Software Inc.
Patient Care Technologies (PtCT)
Paragraph, a division of Vadem
PhatWare Corp.
Physix Inc.
Proxim Inc.
Puma Technology Inc.
Rand Software Corp.
River Run Software Group Inc.
Ruksun Software Technologies Ltd.
Sierra Imaging Inc.
Socket Communications Inc.
Spyglass Inc.
Sybase
Syware Inc.
Teletype Co.
Traveling Software, Inc.
WESTTEK L.L.C.

Applications

Data synchronization.
Packing more power into a small device is not the only advance in technology. Hardware and operating systems are still just platforms for applications. There are many problems that standard applications have running on HCDs, and the problems are not just limited to the HCD. Most applications for the HCD are written as stand-alone applications that occasionally synchronize with a desktop system, exchanging data in a backup situation. What about a network connection? Is it possible to connect occasionally to a network, with an HCD, and have the network treat the HCD as a special node on the network? Due to its attributes, its mobility, power restrictions, and purposes, an HCD is a sporadic network connection, not continuous. There are some problems that arise from this relationship. The network should be able to recognize this device as a sporadic connection, and maintain states, and other connections in spite of its nature. There has to be software in place to minimize the complexity of such connections. For example, when an HCD connects to a database occasionally, then there can be issues of synchronization, as the database server does not keep the HCDs entries consistent in

real time, but only on connection. And the HCD certainly cannot download certain large results, due to memory and bandwidth considerations.

A recent software innovation comes from Oracle. It is interesting that a company of this size, a company that has built its reputation and business on large systems, has moved into this market. On December 8, 1998, Oracle announced a new component for its system, "Oracle8I™ Lite, the mobile component of the Oracle Internet platform designed to eliminate the complexity of mobile computing. (Oracle Corp, 1998)"

According to Oracle's press release:
> "Oracle8i Lite will eliminate the major barriers associated with mobile computing such as the development, maintenance and deployment of separate applications for mobile and networked users. Oracle8i Lite will provide the industry's first development environment for creating thin client applications that work identically on enterprise-systems and mobile devices...Oracle8i Lite is a single-user, 64KB-750KB memory footprint, object-relational database specifically designed for mobile computing applications requiring seamless synchronization with central database servers. It comes with an optional mini Web server, full Java support and is optimized for use on small mobile application clients, such as laptops, handheld computers and personal digital assistants (PDAs). Oracle8i Lite supports Java stored procedures and seamless persistent Java mapping. It also has a native JDBC driver and can take advantage of the easy-to-use SQL/J standard for embedding SQL directly in Java programs. Applications use JDBC or SQL/J to access Oracle8i. (Oracle Corp., 1998)"

Advance Systems

HCD software has been generally written for stand-alone use. The programs will use data synchronization to move data back and forth from desktop host to HCD. These programs include contacts databases, e-mail, small notepad-type applications, small spreadsheets. These programs are not, however, network aware, in that they are not now able to connect to networks and utilize the full power of these networks. Their browsing capabilities are minimal, and they cannot link directly to networks as a node, only to synchronize data. Advance Systems has products to go beyond this limitation.

Advance Systems, a British company, has produced software to give network connectivity to HCDs. An HCD can connect directly to a server running Microsoft Exchange, and perform some of the same e-mail functionality and collaboration tasks that Exchange provides to its desktop connections. ASL-Connect is their package and it provides the following:
- Automatic backup and Restore of Handheld Computers
- Automatic installation and configuration of applications
- Synchronization (Replication) of E-mail, Calendar, Address Book and To-Dos with Lotus Domino Server
- Synchronization of 3rd Party Notes apps and your own Notes apps with handheld computers

- Synchronization of handheld applications with Oracle, DB/2. Sybase + other leading relational database servers
- Comprehensive logging of handheld state and all errors for help desk staff to diagnose problems
- Data integrity and performance features to make the best of unreliable cell phone and wireless links
- Connect using Modem, Cell Phone, Wireless, via a PC, via the Internet or using a network card.
- Scalability - a single processor Pentium II Server can support thousands of users. Add extra processors or servers to support even more users. (Advance Systems, Inc. 1998)

Networking and HCDs

Until HCDs can connect to networks effectively, then they remain quite limited in use. It is the connection to the network that actually delivers the full power of the network to the hand held level. This connection will require not just new programs for the HCDs, but server side intelligence in knowing how to treat these clients. The HCD has added even more heterogeneity to the network, and its integration is the next step in development.

LARGE KNOWLEDGE AND DATA BASES

Introduction

Since the earliest days of computers, data has been stored in many formats. There are unstructured files, structured files, databases of many different types, multimedia files, and now we have an Internet filled with these and more. Moreover, in order to access the potential of this vast knowledge and data system, we have to find ways to select useful data and present it in a coherent and timely fashion.

There are new types of queries to be answered. A single database cannot hold all the answers to all the questions that one could compose, and the world is now asking questions of their data sources that move beyond the boundaries of a single database. The answers to the myriad of potential questions require the integration of multiple data sources.

One substantial thrust of computer research, whether academic, military or commercial, is in the area of large knowledge bases. Researchers are finding ways to integrate heterogeneous data sources, creating large knowledge bases from these disparate sources by creating and implementing logical rules to operate on the data.

Research

Projects

There are many different sources of data, and many different data types within these sources. There are files of data from the 1950's and 1960's, when storage was slow and expensive that are packed such that every bit in the file has a particular meaning, and there is almost no wasted space. And yet there are contemporary DBMSs that utilize the speed and availability of storage to maintain additional data on every piece of data placed into the database. The problem then becomes how a user can easily access knowledge gleaned from the joining of data from these two data sources through a single integrated view.

Old structured and unstructured files and databases actually used very few different datatypes: strings, integers, floats, etc. the new versions of database management systems include all sorts of new native datatypes: video, audio, hypertext links, Java applets, and others. We can also view many of the old structures as datatypes themselves: documents, spreadsheets, programs, etc. These files contain useful data and information, and one of the serious issues of this research is in developing methods to actually investigate old software and documents, extract data, and structure it. We look towards this research to integrate all of these datatypes.

Large data- and knowledge- base integration is another hot topic for researchers. There may be hundreds of projects completed, ongoing and proposed that investigate the many aspects of this problem. The following paragraphs detail some of the latest projects in this arena, software development that proposes to perform this integration.

V.S. Subrahmanian, of the University of Maryland, has completed a project for DARPA that looks at the development of a program he calls a mediator, derived from an idea of Gio Weiderhold (Subrahmanian, 1998). This software, called WebHERMES, uses a mediator language called HERMES (Heterogeneous Reasoning and Mediator System). A mediator is a program that integrates multiple databases, and Subrahmanian goes even further in that the mediator that he proposes will look at other data structures besides databases.

HERMES is a software package consisting of a set of functions that take data sources, which are viewed abstractly as objects, as input. These functions return data objects. There is "a set of relations on the data-objects...[that] may be thought of as the predefined relations in the domain (Subrahmanian, 1998)."

His project has looked at the following:
- Query optimization, where the reasoning process is cached to avoid multiple massive computational tasks.
- Resolving "syntactic and semantic conflicts between (the data in) the disparate data sources."
- Forming and answering "personal" queries, queries specific to the user and his/her needs.
- "Maintaining a mediator against changes to the data sources, in the form of restructuring."
- Security of all the data in the system, from record level security, file security, and security of the integrated information that is returned to the view.
- Web access to the technology
- Addressing some of the issues of heterogeneous multimedia databases, adding some kind of mathematical structure to this type of data and its integration (Subrahmanian, 1998).

A Global Database Management System (GDBMS) is another approach to the problem (Capraro, 1997). This is a functional approach to integrating database management systems (DBMS) and involves creating a database of the metadata of the systems to be integrated. Its advantage is that it is not limited to a set of functions that can run, but queries can be composed, on the fly through accessing the metadata. The metadata can then reflect the changes that the component database schemas undergo without reprogramming these functions. Its disadvantage is that it is limited to databases, and does not encompass structured and unstructured files, multimedia files, programs, and other non-database data sources.

Yigal Arens has developed the SIMS approach to integrating multiple heterogeneous data sources (Arens, 1998). Single Interface to Multiple Sources of Incomplete Data (SIMS) is an development of artificial intelligence research, "primarily in the areas of knowledge representation, planning and machine learning."

SIMS creates a model description of each data source independently, utilizing source analysis software. It then accepts queries, plans the query process, optimizes the process based on learning and performance metrics, and returns the results. Some of the most important contributions are in the areas of knowledge representation and modeling, semantic rule discovery and learning processes based on query returns.

Query Languages

Retrieving data from these many data sources is usually the work of a query language. Query languages are high-level methods of accessing data, usually from databases, but there are some languages that access data from other sources as well. Query language development is an important component in large knowledge base development, as the query language is the representation of the knowledge required by the user.

Loom is a project of the Artificial Intelligence research group at the University of Southern California. It is the query language used by the SIMS project. More than a simple data retrieval language, it is a method of "constructing intelligent applications." Loom contains "definitions, rules, facts, and default rules" that work together in a deductive engine "to compile the declarative knowledge into a network designed to efficiently support on-line deductive query processing (Loom, 1998)." Loom queries can then intelligently move around a network, gathering information from multiple sources.

KQML is another query language that may be quite useful in the large knowledge and data base integration. It is specified in the Intelligent Networking section of this paper. It specifically is used to interact with various intelligent agents, each of which goes out through a network gathering data/information.

SQL, or Structured Query Language, has become the standard query language of the database world. Because it is a standard, its syntax can be used to communicate across databases, allowing programs to access several databases utilizing the same basic syntax. SQL has been around since the early 1970's, growing out of the original research work in relational databases. Oracle used SQL as its query language starting in 1979. Standards were published in 1986, and with the standards, its use has become the standard of the database industry.

The problem with standards in general is that they tend to lag behind software innovations. There have been several additions to the ANSI SQL standard, including a major revision to the entire language in 1992, and a revision to the latest proposed standard, SQL3, has been finalized late in 1998. This new standard, which incorporates many new features to SQL, and also adds object references and object features such as encapsulation, subtypes, inheritance, and polymorphism.

OQL, or Object Query Language, is a development rising from SQL. It is a query language that operates on objects, not just data. It incorporates some of the standard properties of objects such as polymorphism and methods that can modify data within the query. Its implementation is quite limited as of now, but even if it does not survive in this form, its concept may be quite important given the rise of object-oriented technology through computer science.

ODBC, Open Data Base Connectivity, is Microsoft's solution to connecting databases. Because of Microsoft's market share, its standard, published in 1994, has quickly become the industry standard. ODBC is a connectivity program and standard for Windows-based computers that

allows connectivity to compliant databases through the operating system. ODBC allows SQL statements to run across the operating system to many different database management systems. It is the most important connectivity standard in Windows-based database integration.

Other Research Issues

Much of the work in creating and managing large knowledge bases is being done with data that already exists in DBMS-managed systems, or at least in structured files of some type, such as spreadsheets. The next major issue is that of unstructured data. There is a lot of valuable information and data that exists in unstructured text files, HTML pages, and within programs themselves.

Applications

The field of large knowledge and data bases is not limited to research alone. There are many different commercial and government software packages that integrate data from various data sources. The following packages are examples of a market that is growing.

Broadsword

Project Broadsword is a DoD effort aimed at integrating many data sources on the analyst's desktop. Utilizing a 3 layer architecture, Broadsword takes raw input from a user and returns data and information from many different data sources. It utilizes data plug-ins to access these data sources. By using different plug-ins for different data sources, Broadsword is able to retrieve data and information in a consistent manner. The display does not integrate the data with SQL joins or other integrating techniques, but it does allow the user to view each piece of the data retrieved.

The purpose of Broadsword is to allow a user to retrieve data and information from different sources without having to know where or what those sources are. The user runs queries to Broadsword, which retrieves data from each source specified, and displays each result as required. It uses the concept of a "Librarian" that knows where the data exists, and how to get it to the user. In addition, Broadsword integrates security throughout the system.

Oracle

The latest version of the Oracle database, version 8*i* (the "*i*" stands for Internet) incorporates many database integration tools. Oracle has been at the forefront of commercial database development since the 1970's, and it holds tremendous market share in the large commercial database market.

One of the enhancements this version offers is the ability to automatically publish data on the Internet as HTML. Oracle builds HTML from data pieces, and because it is built, it can be accessed again as data in the database, and not just free text in HTML. As more and more

Internet sites are built with tools like this, that is, publishing pages from databases, the data displayed is actually structured, and therefore accessible to database integration and connectivity.

The large database systems have increased the number of datatypes that are native to their systems. Oracle actually allows Java applets to be stored within a database. This means that data and information that may exist within a program itself may be accessible through the DBMS.

Another important part of the Oracle product is the Designer's kit that allows reverse engineering and modeling of existing databases. This functionality is not limited to Oracle, as many designer tools contain this ability. This product, and others like it, can add to database integration by modeling the database, allowing developers to logically integrate the disparate sources within applications.

Brann

Brann Software, a British firm, has developed some software products that attacks some of the problems of integrating unstructured and non-database data. Brann Viper is a tool that analyzes unstructured data to allow it to be incorporated into a database. In addition to database query techniques, this product uses pattern recognition, neural nets, OLAP, rule induction, and statistics to search documents to answer real language queries. The data acquired by Viper can then be exported to true database records for subsequent structured retrieval.

Brann Asp is another Brann product:
> "Brann Asp is a middleware product that has been developed to ease the process of collating, cleaning and structuring data for analysis. It is a "data refinery" that allows users to extract data from many different sources, such as relational databases, bought-in lifestyle data, campaign responses, accounts systems or literally any "data rich" environment, and to transpose it into a format that can be handled by products such as Brann Viper. (Brann, 1998)"

Products such as these are on the market right now, integrating heterogeneous data sources.

SUMMARY

Capraro Technologies, Inc. has researched the current state of technology in the areas of intelligent agents, networking and the Internet, hand held computing devices, and large knowledge bases and databases. This report details the results of that research. One important attribute of this research is that each of these areas is changing rapidly, with new research and products arriving almost daily. We need to recognize that this technology, because it is popular both as products and research, will continue to progress, requiring a continuing effort on the part of interested parties in keeping up with the next changes.

We intend to contribute to the technology changes in our own implementation. As part of this project, we will provide a demonstration of the coordinated use of these technologies to provide an efficient means of intelligent networking, and to demonstrate its feasibility. Our demonstration will utilize a handheld device to connect to a network, run queries on multiple databases, join and provide the results of these queries on the device. We will utilize an intelligent networking solution to determine what type of device submits the query, and then scale the results based on the parameters of this device. Given a query that returns a large amount of data, we should be able to determine the capacity of the device to process these results and scale the returns to the device. An HCD will be served with appropriate data/information based on its hardware and bandwidth limitations. If this same query is called from a large, powerful workstation, it should be served with more of the resultant data/information, because the network will be able to determine the ability of each machine to download, process and display these returns.

REFERENCES

Agent News WebLetter 3.02, http://www.cs.umbc.edu/agents/agentnews/

Bradshaw, J. 1997. Introduction to Software Agents, in Software Agents, ed. J. Bradshaw, Cambridge:AAAI Press/The MIT Press, 3-46.

Brann Software. 1998 http://www.brann.co.uk/

Grimshaw, A.S., Wulf, W.A., & the Legion Team. 1997.The Legion vision of a worldwide virtual computer. Communications of the ACM, 40(1) 39-45.

Hermans, B. 1997 Intelligent Software Agents on the Internet: An Inventory of Currently Offered Functionality in the Information Society and a Prediction of (Near) Future Developments. In First Monday, http://www.firstmonday.dk/issues/issue2_3/ch_123/

IQ Software, http://www.iqsc.com/products/family.htm

Jain, R. & Ramakrishnan, K.K., 1998 Congestion Avoidance In Computer Networks With A Connectionless Network Layer Part I: Concepts, Goals And Methodology, Digital Equipment Corporation.

Jain, R., 1992 "Myths about Congestion Management in High Speed Networks," Internetworking: Research and Experience, 3, pp. 101-113

Kay, A. 1984. Computer Software. Scientific American 251(3): 53–59.

Labrou, Y. & Finin, T., Semantics and Conversations for an Agent Communication Language, Sept 18, 1998, (This work was supported in part by the Air Force Office of Scientific Research under contract F49620{92-J-0174, and the Advanced Research Projects Agency monitored under USAF contracts F30602-93-C-0177 and F30602-93-C{0028 by Rome Laboratory.)
Nardi, B. A.; Miller, J. R., and Wright, D. J. 1998. Collaborative, Programmable Intelligent Agents, Communications of the ACM, 41(3), 96-104.

Paepcke, A., et.al., 1998 Using Distributed Objects for Digital Library Interoperability, Stanford University http://computer.org/computer/dli/r50061/r50061.htm

Parunak, H. V. D. 1998. Practical and Industrial Applications of Agent-Based Systems. Industrial Technology Institute.

Quantum Flow Control Corp. 1998 http://www.qfc.org

Sage Research, Inc., 1998 Quantitative Analysis of the ATM LAN Market, Prepared for the ATM Forum, http://www.atmforum.com/atmforum/library/sage_presentation/tsld001.html

Shoham, Y. 1997. An Overview of Agent-oriented Programming. In Software Agents, ed J. M. Bradshaw. Menlo Park, Calif.: AAAI Press.

University of Wisconsin, Computer Science Department, 1998. Overview of the Condor High Throughput Computing System. http://www.cs.wisc.edu/condor/overview/

Wooldridge, M. J., and Jennings, N. R. 1995. Agent Theories, Architectures, and Languages: A Survey. In Intelligent Agents: ECAI-94 Workshop on Agent Theories, Architectures, and Languages, eds. M. J. Wooldridge and N. R. Jennings, 1–39. Berlin: Springer-Verlag.

Wooldridge, M. J., and Jennings, N. R. 1998. Applications Of Intelligent Agents. In Agent Technology Foundations, Applications, and Markets, eds. M. J. Wooldridge and N. R. Jennings, Berlin: Springer-Verlag.

APPENDIX A.

Handheld Computing Devices Comparison Charts

The following charts list most of the available handheld computing devices, and list some of their specifications. For those machines that can utilize PCM/CIA cards, the capabilities of the machines can be increased dramatically. There are memory cards for as much as 220 MB, and CompactFlash memory of up to 48MB.

Mfg. & Model	Weight (oz)	Processor	Max ROM/RAM (MB)
3Com Palm III	6	Motorola DragonBall 68328	2/2
Casio Cassiopeia A-20	16.3	Hitachi SH-3 (80 MHz)	8/8
Compaq C-Series 2015c	17	High-speed MIPS-based RISC	16/32
Datarover 840	18.4	MIPS R3000 RISC	8/4
Everex Freestyle Executive A-20	5.3	NEC VR4111 (66 MHz)	8/16
Geofox-One Professional	13.7	ARM-7 CL-PS7110 (18 MHz)	8/16
HP 360LX	15.6	Hitachi CPU SH3-based (44 MHz)	10/8
HP 660LX	20.6	Hitachi SH3	12/32
Hitachi HPW-200EC	20.6	SuperH SH-3 (100 MHz) RISC	16/48
IBM WorkPad	5.7	Motorola DragonBall 68328	1/1
LG Phenom Ultra	29.9	Hitachi SH3 (100 MHz)	12/32
NEC MobilePro 700	23.7	NEC Vr4102	16/8
NEC MobilePro 750C	29.8	NEC Vr4111 (80-MHz MIPS)	16/32
Novatel Wireless Contact	21.8	Hitachi SH3 (80 MHz)	16/32
Philips Nino 312	7.8	Philips R3910-32 bit	32/8
Philips Velo 500	15.3	Philips PR31700	32/24
Psion Series 5	12.5	RISC ARM 7100	6/10
Sharp Mobilon HC-4100	14.1	MIPS RISC processor	12/32
Sharp Mobilon HC-4500	14.1	MIPS RISC processor	12/32
Sharp Mobilon Pro 5000	41.6	MIPS RISC Processor	16/16
Sharp SE-500 Mobile Organizer	7	Sharp proprietary	1/1
Sharp Zaurus ZR-3500X	12	Sharp proprietary	1/1
Texas Instruments Avigo 10	7	Custom Z-80	2/2
Uniden UniPro 100A		Phillips 31700 (75Mhz)	4/8

Mfg. & Model	Dimensions (WxDxH) (in.)	Key-board	Stylus	Hand Writing Recog.	Display Size (in)
3Com Palm III	3.3x4.7x.6	No	Yes	Yes	2.5x3.25
Casio Cassiopeia A-20	7.3x3.8x1	Yes	Yes	No	2.3x6.3
Compaq C-Series 2015c	7.3x3.9x1.6	Yes	Yes	No	6.1x2.3
Datarover 840	7.3x4.6x1.3	No	Yes	Yes	
Everex Freestyle Executive A-20	4.8x3.2x.7	No	Yes	No	3.1x2.4
Geofox-One Professional	7.4x4.7x 75	Yes	No	No	3.2x6.1
HP 360LX	7.2x3.7x1.1	Yes	Yes	No	6.25x2.5
HP 660LX	7.8x4.1x1.4	Yes	Yes	No	6.25x2.5
Hitachi HPW-200EC	10x5.2x1.3	Yes	Yes	No	8.1 diag
IBM WorkPad	3.1x4.6x.6	No	Yes	Yes	2.5x3.3
LG Phenom Ultra	9.96x5.2x1.3	Yes	Yes	No	7.5x3
NEC MobilePro 700	9.4x4.7x1.1	Yes	Yes	No	2.75x6.9
NEC MobilePro 750C	9.6x5.4x1.3	Yes	Yes	Yes	7.5x3
Novatel Wireless Contact	7.6x4.9x1.3	Yes	Yes	No	6.75x2.75
Philips Nino 312	5.25x3.4x.87	No	Yes	No	2.4x3.2
Philips Velo 500	6.7x3.7x.94	Yes	Yes	No	2.3x5.8
Psion Series 5	3.5x6.7x0.9	Yes	Yes	No	5.3x2
Sharp Mobilon HC-4100	7.1x3.6x1.1	Yes	Yes	No	6.1x2.4
Sharp Mobilon HC-4500	7.3x3.7x1.2	Yes	Yes	No	6.1x2.4
Sharp Mobilon Pro 5000	9.3x7.9x1.0	Yes	Yes	No	8.2 diag
Sharp SE-500 Mobile Organizer	5.9x3.5x0.7	No	Yes	No	3.8x2.3
Sharp Zaurus ZR-3500X	6.3x6.7x0.9	Yes	Yes	No	3.5x2.5
Texas Instruments Avigo 10	3.25x5.50x.75	No	Yes	No	2x3
Uniden UniPro 100A		No	Yes	Yes	

All of these devices have backlit screens.

Mfg. & Model	Resolution (pixels)	Colors	Touch Screen	Type II Slots	Modem
3Com Palm III	160x160	4 Gray	Yes	0	optional (14.4 kbps)
Casio Cassiopeia A-20	640x240	4 Gray	Yes	1	PCMCIA
Compaq C-Series 2015c	640x240	256	Yes	1	33.6 KBPS
Datarover 840	480x320	4 Gray	Yes	2	19.2 KBPS
Everex Freestyle Executive A-20	240x320	4 Gray	Yes	0	33.6 KBPS
Geofox-One Professional	640x320	16 Gray	No	1	33.6 KBPS
HP 360LX	640x240	4 Gray	Yes	1	PCMCIA
HP 660LX	640x240	256	Yes	1	56.6 KBPS
Hitachi HPW-200EC	640x240	256	Yes	1	33.6 KBPS
IBM WorkPad	160x160	4 Gray	Yes	0	Optional
LG Phenom Ultra	640x240	256	Yes	1	33.6 KBPS
NEC MobilePro 700	640x240	4 Gray	Yes	1	33.6 KBPS
NEC MobilePro 750C	640x240	256	Yes	1	33.6 KBPS
Novatel Wireless Contact	640x240	16 Gray	Yes	1	(wireless and wireline)
Philips Nino 312	320x240	4 Gray	Yes	0	28.8 kbps
Philips Velo 500	640x240	16 Gray	Yes	0	28.8 kbps
Psion Series 5	640x240	16 Gray	Yes	1	Optional
Sharp Mobilon HC-4100	640x240	16 Gray	Yes	1	33.6 KBPS
Sharp Mobilon HC-4500	640x240	256	Yes	1	33.6 KBPS
Sharp Mobilon Pro 5000	640x480	4096	Yes	1	33.6 KBPS
Sharp SE-500 Mobile Organizer	240x159	4 Gray	Yes	0	14.4 kbps
Sharp Zaurus ZR-3500X	320x240	4 Gray	Yes	0	14.4 kbps
Texas Instruments Avigo 10	240x160	4 Gray	Yes	1	None
Uniden UniPro 100A	320x240	4 Gray	Yes	0	28.8 kbps

Mfg. & Model	Other Ports	Battery Life	Audio Speaker	Micro-phone
3Com Palm III	None	25 hrs.	Yes	Yes
Casio Cassiopeia A-20	CompactFlash	25 hrs.	Yes	Yes
Compaq C-Series 2015c	IrDA	3 hrs.	Yes	Yes
Datarover 840	IrDA	8 hrs.	Yes	Yes
Everex Freestyle Executive A-20	IrDA, CompactFlash	7-8 hrs.	Yes	Yes
Geofox-One Professional	None	25 hrs.	Yes	Yes
HP 360LX	CompactFlash	15-20 hrs.	Yes	No
HP 660LX	CompactFlash	4-6 hrs.	Yes	No
Hitachi HPW-200EC	IrDA, VGA-out	8 hrs.	Yes	Yes
IBM WorkPad	None	1,344 hrs.	Yes	No
LG Phenom Ultra	IrDA, VGA-out, CompactFlash	8 hrs.	Yes	Yes
NEC MobilePro 700	VGA	25 hrs.	Yes	Yes
NEC MobilePro 750C	VGA-out, CompactFlash	8 hrs.	Yes	Yes
Novatel Wireless Contact	IrDA, VGA-out, CompactFlash	12 hrs.	Yes	Yes
Philips Nino 312	IrDA, CompactFlash	10-12 hrs.	Yes	Yes
Philips Velo 500	IrDA	15 hrs.	Yes	Yes
Psion Series 5	CompactFlash	35 hrs.	Yes	Yes
Sharp Mobilon HC-4100	IrDA	25 hrs.	Yes	Yes
Sharp Mobilon HC-4500	IrDA	3-6 hrs.	Yes	Yes
Sharp Mobilon Pro 5000	IrDA	15 hrs.	Yes	Yes
Sharp SE-500 Mobile Organizer	IrDA	100 hrs.	No	No
Sharp Zaurus ZR-3500X	IrDA	100 hrs.	No	No
Texas Instruments Avigo 10	IrDA	2-3 mo.	No	No
Uniden UniPro 100A	IrDA, CompactFlash	15 hrs.	Yes	Yes

Mfg. & Model	Serial Ports	Audio Inputs	Audio Outputs	Operating System
3Com Palm III	1	0	0	Palm OS 3.0
Casio Cassiopeia A-20	1	0	0	Windows CE 2.0
Compaq C-Series 2015c	1	0	1	Windows CE 2.0
Datarover 840	1	1	1	Magic Cap 3.1
Everex Freestyle Executive A-20	1	1	1	Windows CE 2.0
Geofox-One Professional	1	0	0	Psion EPOC32
HP 360LX	1	0	0	Microsoft CE 1.0
HP 660LX	1	0	0	Windows CE 2.0
Hitachi HPW-200EC	1	0	0	Windows CE 2.0
IBM WorkPad	0	0	0	Palm OS 3.0
LG Phenom Ultra	1	1	1	Windows CE 2.0
NEC MobilePro 700	1	0	0	Windows CE 2.0
NEC MobilePro 750C	1	0	0	Windows CE 2.0
Novatel Wireless Contact	1	1	1	Windows CE 2.0
Philips Nino 312	1	0	0	Windows CE 2.0
Philips Velo 500	1	0	0	Windows CE 2.0
Psion Series 5	1	1	1	Symbian EPOC32
Sharp Mobilon HC-4100	1	0	0	Windows CE 2.0
Sharp Mobilon HC-4500	1	0	0	Windows CE 2.0
Sharp Mobilon Pro 5000	1	0	0	Windows CE 2.0
Sharp SE-500 Mobile Organizer	1	0	0	Sharp proprietary
Sharp Zaurus ZR-3500X	1	0	0	Sharp Synergy
Texas Instruments Avigo 10	0	0	0	proprietary
Uniden UniPro 100A	1	0	0	Windows CE 2.0

Much of this data was found at the CNET, Inc. web site. The URL is found in the references section of this paper.