Generating Command and Control Systems in an Hour: The Microsoft Way

Alistair Barros[1], Charles Herring[2], John Hildebrandt[3] and Michael Rees[4]

DSTC Pty Ltd [1]
University of Queensland
Level 7, GP South
Brisbane QLD 4072
Australia
Phone: (07) 33658358
Fax: (07) 33651999
E-mail: abarros@dstc.edu.au

University of Queensland [2]
Department of Computer Science and Electrical Engineering
Level 6, GP South
Brisbane, QLD 4072
Australia
Phone: (07) 3365 2903
Fax: (07) 3365 1999
Email: herring@dstc.edu.au

DSTO C3 Research Centre [3]
Fernhill Park
Department of Defence
Canberra, ACT 2600
Australia
Phone: (02) 6265 8002
Fax: (02) 6265 8010
Email: john.hildebrandt@dsto.defence.gov.au

Bond University [4]
School of Information Technology
Gold Coast, QLD 4229
Australia
Phone: (07) 5595 3351
Fax: (07) 5595 3320
Email: mrees@bond.edu.au

Abstract

In this paper the authors investigate the benefits of adopting single vendor architectures to develop C2 applications. The component based architecture chosen for the exercise was the Microsoft Windows 2000 DNA Architecture. While single vendor architectures have disadvantages and risks it was felt that the advantages of the approach also needed investigation as input into unbiased discussion on the development of future C2 systems. Current and past approaches to C2 system development based on closed and supposedly open

approaches have had a variety of problems. It is felt that the focus in the future needs to move away from building systems and move towards the dynamic, rapid and flexible integration of existing technologies. In this paper we describe a number of experiments and studies aimed at developing components of functionality useful in C2 systems rapidly using off the shelf components from a single commercial architecture.

1 Introduction

Development of Command and Control (C2) systems has been characterised by very large software development projects. These projects have typically been extremely expensive, late to deliver and slow to adapt to user needs. To address these issues software development and project management processes have evolved but problems still exist. The DOD's Global Command and Control System (GCCS) is an excellent example illustrating the multitude of "challenges" facing traditional C2 systems development efforts [GAO, 1998].

In the commercial world increasing software development costs, the pressures of electronic transactions and user expectations of Internet access are driving business to increase their reliance on commercial of the shelf (COTS) software solutions. This allows businesses to leverage the development efforts of large COTS vendors, customising and integrating the products into their business systems. Hence rather than the effort being dominated by large in-house software projects the development process consists of discovery of COTS components that can deliver the user requirements followed by integration of the components and development of specialised functionality if required.

While C2 systems have adopted COTS products more widely the use of products from a range of vendors on diverse platforms and architectures has created an integration burden for the C2 projects. This has resulted in the requirement to develop large amounts of 'glue' code and maintain a rigid configuration of COTS products. Such an environment limits the speed at which new technology can be introduced and large bodies of glue code become significant development efforts in their own right. In an attempt to mitigate just some of these problems the DOD has created the vast bureaucratic effort known as the Defense Information Infrastructure Common Operating Environment (DII COE) [DII, 2000].

In this paper we offer a solution to the problems of building C2 systems described above. The first step is to change our perspective from building systems to providing users the means to generate their own C2 systems. We believe the day of building gargantuan C2 software systems has ended. Instead, we should be aiming to provide commanders and their staffs with "C2 Wizards" that, based on answering 20 questions, will generate a command and control and/or command support system tailored for the mission at hand. The second step, that makes the first possible, is to reduce the DII COE to one word: Microsoft. Then our "C4ISR Architecture" becomes the Microsoft Windows 2000 DNA [MS-DNA]. This allows multiple products from multiple vendors to be integrated in a consistent manner. While each individual Microsoft technology may not be the best of breed, the ease of integration between the separate products delivers a powerful and flexible platform for rapidly constructing customized solutions. In fact as noted in [Sprott, 2000] enterprises which attempt to choose "best of breed" from diverse vendors architectures can find getting the pieces to work together is a major effort and the effective functionality of each piece can be reduced by the requirement to stick to common denominators betweens the components.

We support the position taken in this paper based on a series of technology experiments conducted over the last year aimed at better understanding the Windows 2000 DNA architecture. These experiments demonstrate that C2 functionality can be rapidly developed with minimal custom code. We developed applications to illustrate the use of each layer of the DNA n-tier architecture in the context of command and control systems. The applications range from the desktop such as Office 2000, to collaboration support based on IIS/OSE and backend database management systems for image analysis.

2 60-Minute Admin/Logistics Work Group Web

As stated earlier, the approach of building large, custom command support and command and control systems has probably reached its zenith. We think the focus should now be on providing users the flexibility to generate their own systems. Our rationale is based on the following considerations. Firstly, today each operational situation faced by a commander is unique. His requirements to organize and manage that situation cannot have been foreseen by software development efforts begun a decade earlier. Secondly, technology now exists to permit the type of flexibility we think is needed. In particular, the Internet and related technologies that were originally developed by the US DOD for command and control are now commercially available. Also, and very importantly, the level of computer literacy of military staff personal is high. As an example of this approach, we used COTS software to develop an Admin/Logistics workgroup web site. Then, to demonstrate the versatility of this approach, we built a workgroup web housing a repository of documents for health intelligence analysts.

This experiment demonstrates how staff officers can generate a collaborative command support system in an hour for a variety of purposes. During an interview with a staff logistics planner the following question was asked: “How do you do run your shop?” He went to a whiteboard and in about twenty minutes sketched his major areas of concerned, the types of projects, the people and their roles that worked under him, the software tools he used (many he had developed in Excel), etc. We then used the “Microsoft 60 Minute Intranet Kit” Wizard [MS-KIT] in FrontPage 2000 to generate a collaborative support tool for administrative and logistics planning, and then for health intelligence operations. Each support tool is an interactive workgroup web site shared by the personnel involved in a team.

The Wizard asks the following sequence of questions:

1. Tell us about you company: company name, title for the web site and contact email address.
2. How do you want to organize document storage: by project or department. We chose to organize by project, so the wizard prompted us to enter a name for each project.
3. Enter the names/offices for all people that will be members of the workgroup.

That is all the input required. FrontPage then generates the workgroup web site. The home page for administrative and logistics-planning tool is shown in Figure 1.

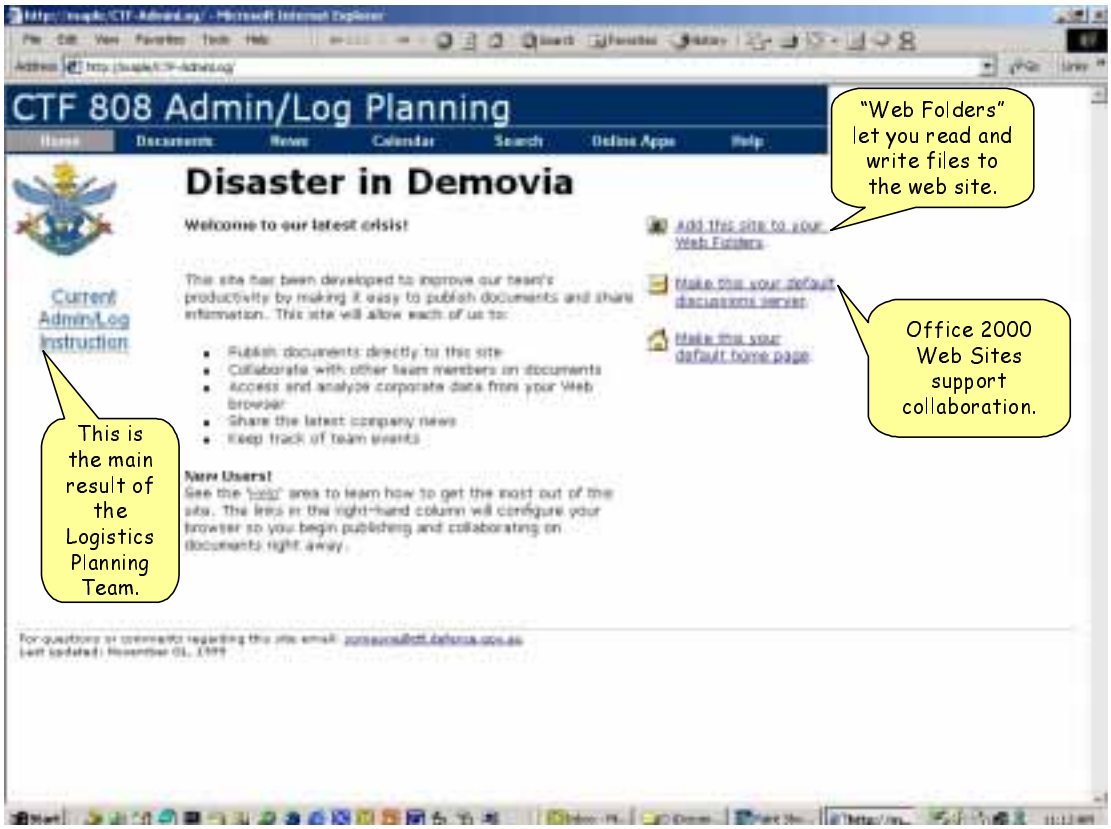

Figure 1 Admin/Logistics Work Group Web Site

We remark on some of the most important features of the workgroup web sites and Office 2000 [MS-O2K] related technologies. Office 2000 provides a set of Office Server Extensions [MS-OSE] that are installed on the machine that hosts the web site. (Microsoft's web server software is called Internet Information Service and abbreviated IIS/OSE when Office Server Extensions are installed.) On the home page is a link to the "Current Admin/Log Instruction." This is the main output of the Admin/Logistics team and is used by subordinate organizations in developing their operations plans. It is a Word document. Across the top of the page is a menu for the major sub-areas of the site: documents (organized by project and personnel), news, calendar, search, on-line applications and help.

Two other features are indicated by call-outs in the figure: web folders and discussions. Web folders permit users to access folders on the web site as if they were local to the user's workstation hard drives. That is, these folders appear on the users local machine like any other folder and the user can manipulate files with the familiar Windows Explorer user interface. The other feature is *discussions*, which is new to Office 2000. Discussions permit users to enter remarks or notes about any item displayed in Internet Explorer (IE) or any Office application. Users can also insert discussion items *inside* any document stored in web page format. For example, users can view the "Current Admin/Log Instruction" in IE and make comments about the document or insert comments about specific paragraphs or sections. They can also use discussions in Word or any other office applications. The discussions are stored on the web site and can be viewed by other users.

The workgroup web site includes areas for each major project. In the case of logistics the projects chosen where: distribution logistics, movement tables, personnel, health-medical, medical etc. Each office/person has a web area for organising documents. These are

essentially home pages with support for organizing work related documents. There is an overall team calendar (generated from Outlook), a news publishing service, and an area for on-line applications for special tools such as movement calculations, a search facility and help. Also, Office 2000 discussions support notification of changes to folders or documents on the web site. Users can subscribe to be notified by email of changes based on a number of criteria.

The "Microsoft 60 Minute Intranet Kit" wizard in FrontPage 2000 was made for general-purpose use. However, with some customisation of the generated web pages the web site could be used to provide a collaborative Admin/Log planning command support tool. The goal of this exercise was to explore the concept of generating command and control tools and systems tailored to the users needs within a short period of time. Development of "C2 Wizards" for each staff function and for specialized planning and execution tasks would be relatively easy using FrontPage and associated Office 2000 tools.

3 Workgroup Web for Health Intelligence Analysts

To demonstrate the flexibility of the workgroup web wizard, we used the same approach to generate a support tool for health intelligence analysts. The requirements were drawn from the HINTS (Health Intelligence System) project. Once again, we used the "Microsoft 60 Minute Intranet Kit" wizard within FrontPage 2000, and nominated projects for each important disease, and user folders for each member of the intelligence team. The home page is very similar to Figure 1.

The HINTS project is based upon a collection of health report and information documents and related document processing tools. The HINTS workgroup web can support many of the requirements of the HINTS project, namely:

1. Folders within the web site store the intelligence documents in a variety of Office formats, HTML and plain text. Thousands of documents can be accommodated in a single web site, or perhaps spread across a number of sites.
2. Collaboration between intelligence analysts is supported using discussions and notifications as described for the Admin/Log planning workgroup web.
3. First level search facilities are provided by the Microsoft Index Server and related ActiveX components.
4. Online applications such as news items, time sheets, and personnel directories provide project management support.
5. Utilising the digital dashboard outlined in the next section, the specialist document analysis applications can be executed and their results displayed.

In this experiment, we categorised the health intelligence documents by disease projects. A typical disease project folder contains related documents as shown in Figure 2.

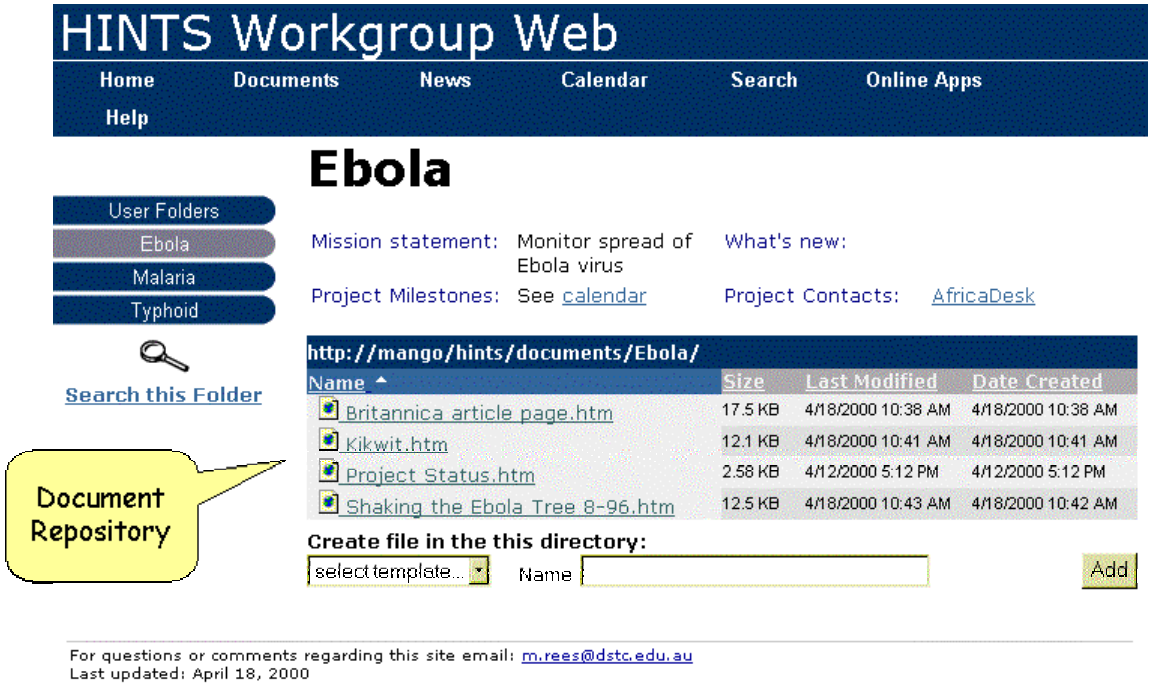

Figure 2. Health Intelligence Documents Folder List.

In Figure 2, the list of documents is automatically generated using an ActiveX control. This allows new documents to be inserted at any time using the web folders mechanism, and the web page will update automatically. From the web page itself, a new folder hierarchy can be created and traversed. In addition, new template files can created from a template list, and edited in-situ from within the web browser.

The Microsoft Index Server is a standard part of the IIS/OSE web server bundle, and can be used to index all files on the server machine. More usefully in the context of our experiment, we used the Index Server to index only the workgroup web sites. This provides the HINTS workgroup web with a powerful search engine for all documents in the repository.

Figure 3. Index Server Search Results.

The Index Server can be configured to index special properties for each document, and these user-specified properties can be search using the Advanced Search capabilities generated by the web site wizard, again shown in Figure 3.

Some of the more powerful features provided by Office 2000 is the ability to insert active Office document fragments within web pages. The workgroup web wizard generates a number of pages containing slide presentations, spreadsheets, charts and pivot tables. All of these fragments play significant roles in presenting live information in C2 scenarios. We exploited live spreadsheets and charts in the HINTS workgroup web for showing some documents statistics as shown in Figure 4.

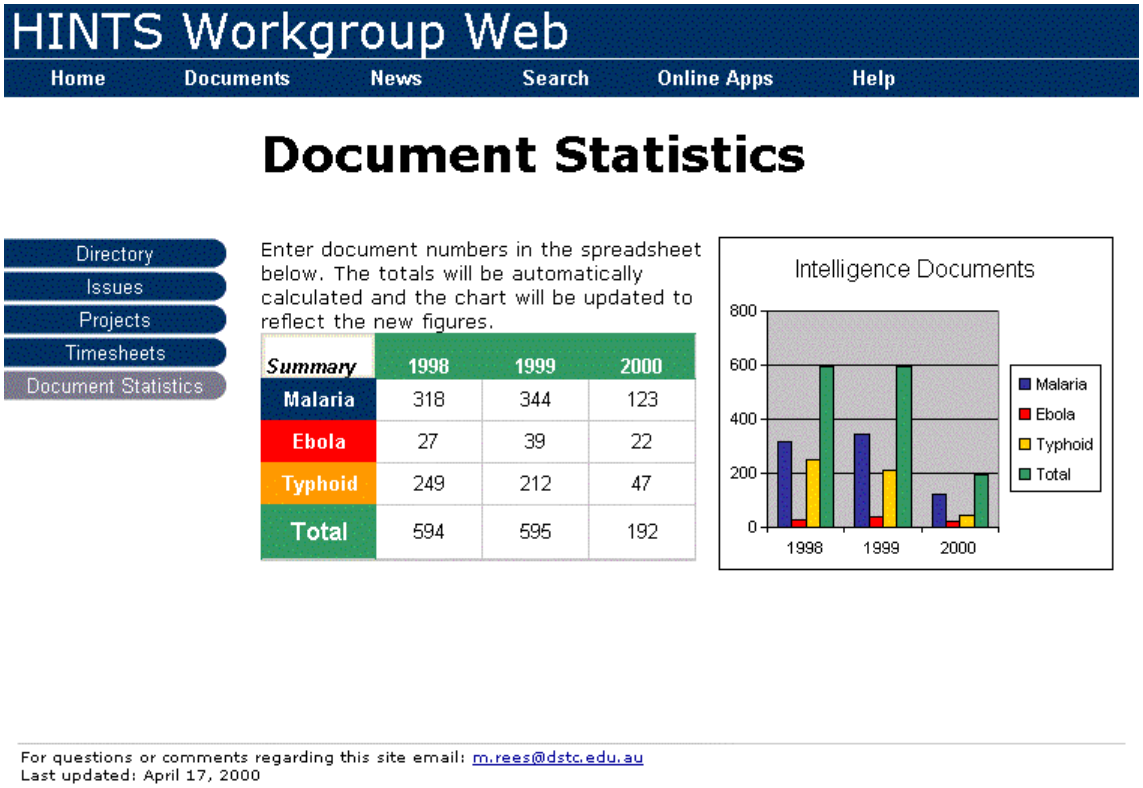

Figure 4. Document Statistics.

When team members view the page in Figure 4 with a browser, they can update values in the spreadsheet within the web page, and see the results updated in the chart. This functionality used the Microsoft Office 2000 Web Components. These are composed of a series of ActiveX controls downloaded on demand.

Once again we showed that very little customisation is needed to generate a fully functional workgroup web for a collaborative team of health intelligent analysts. The Microsoft COTS tools provide the features as standard, and only a little editing with FrontPage 2000 is needed. From that point the team use the web browser and web folders to do its work.

4 Commander's Digital Dashboard

Perhaps the highest level of functionality a command and control system can provide the commander or team leader is an adequate situation awareness report for decision-making. Situation awareness is the timely and accurate view into a complex operation and, to a large

degree, depends on *limiting* the amount of information delivered to commanders. Currently, the commander's staff accomplishes this function. They gather, analyse and filter the glut of information available to the commander and present it to him or her in an organized manner. Naturally we looked to exploit desktop applications to support the commander's situation awareness and decision-making. Here we report on extensions to the experiment using Microsoft Office 2000 related tools.

The following question was asked of a senior staff officer: "What does the general like to see in their daily staff briefing?" (The general in question was in the midst of a major peacekeeping operation at the time and the staff officer knew all too well what he wanted to see. In fact, the officer remarked that it was a full time job to develop the twice-daily briefings in addition to his other full time jobs.) The staff officer's answer: "Traffic Lights." That is, the general wanted to see the status of the major items he monitored marked by coloured dots as Green (good), Amber (iffy) and Red (not good). Each staff officer developed a set of briefings, using PowerPoint, for his particular area of responsibility. For each major item of interest there would be a series of backup slides in case the general asked for more detailed information. At the highest level the general monitored the following items or areas:

- Operations
- Order of Battle
- Logistics
- Personnel
- Communications
- Intelligence.

There are a number of items under each of these categories and these items can be further "drilled down" to get more detailed information.

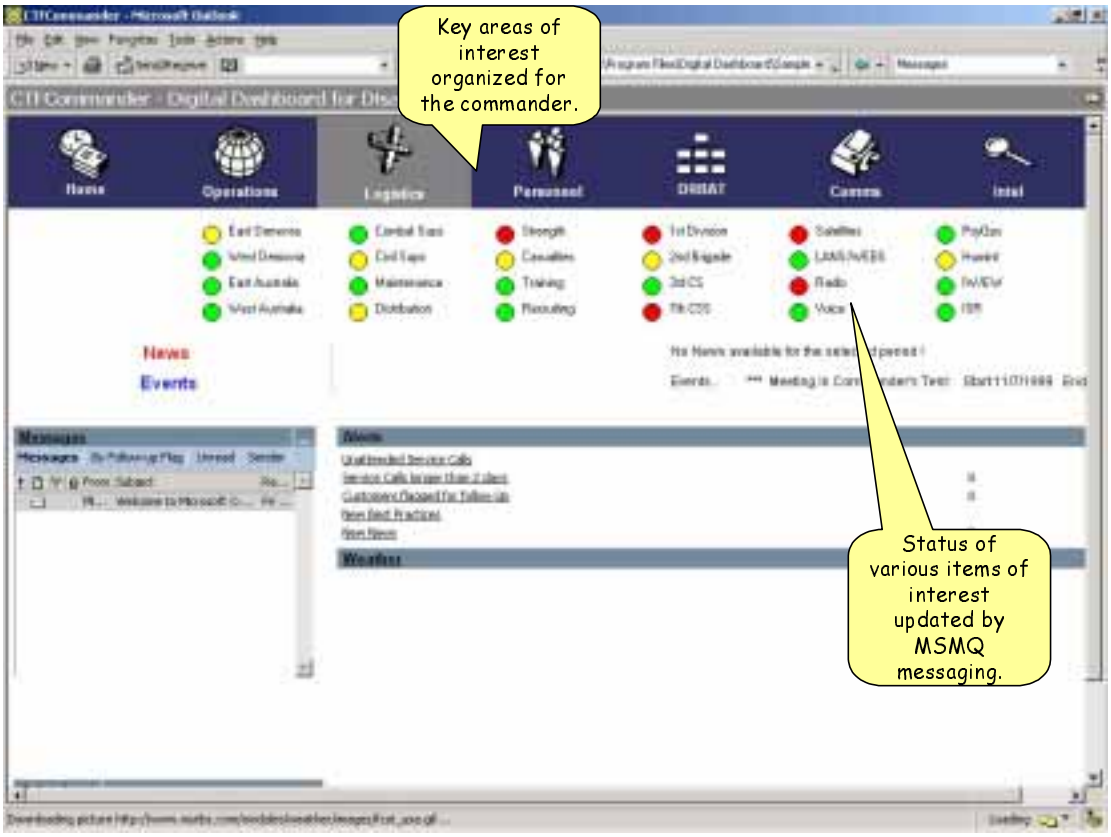

Figure 5. Commander's Digital Dashboard.

We chose to explore the use of Microsoft's "Digital Dashboard" kit [MS-DD] to build an Outlook 2000-based "Commander's Digital Dashboard" for the general. The Digital Dashboard is a software kit and methodology for organizing personal, team, corporate, and external data into the familiar environment of Outlook. This approach provides integrated messaging, calendar and contact management as well as access to any available real-time "Enterprise" information in one application.

In essence, a Digital Dashboard is a dynamic web page backed up by Microsoft Outlook. The term "information nugget" is used to describe the filtered and dynamic information displayed in a Digital Dashboard. The information can be displayed as an ActiveX control, an ASP page, an XML data feed, and so on. The main idea is that a nugget presents information in a concise fashion, focusing on what the user needs, and it is interactive, so the user can drill down into greater detail.

We used the Digital Dashboard kit to develop a "situation awareness" application for the commander. The result is shown in Figure 5. Across the top is a menu with selections for the major areas of interest: Operations, Logistics, etc. By clicking on these menu items the user is taken to more detailed information, for example, the Admin/Logistics workgroup web site shown in Figure 1. Below this menu are the "Traffic Lights" corresponding to each area. We built an ActiveX "TrafficLight" control nugget using Visual Basic. This ActiveX control receives status updates via Microsoft's Message Queue (MSMQ) to provide the commander with instant status information. Each of the traffic lights can have its own unique message queue and can be updated by any number of transport mechanisms (HTTP, SMTP, SOAP, etc.) within the overall Windows DNA architecture. Further, this control can be embedded in any office document such as Word or PowerPoint and receive updates dynamically. Also

shown in Figure 5 are scrolling “ticker tapes” for events and news that can come from the Outlook calendar, Exchange or other sources.

Of course, the main Outlook features of mail, calendar, contacts, and so on, are available in this integrated environment. This is demonstrated when we built a digital dashboard for the HINTS workgroup web as shown in Figure 6.

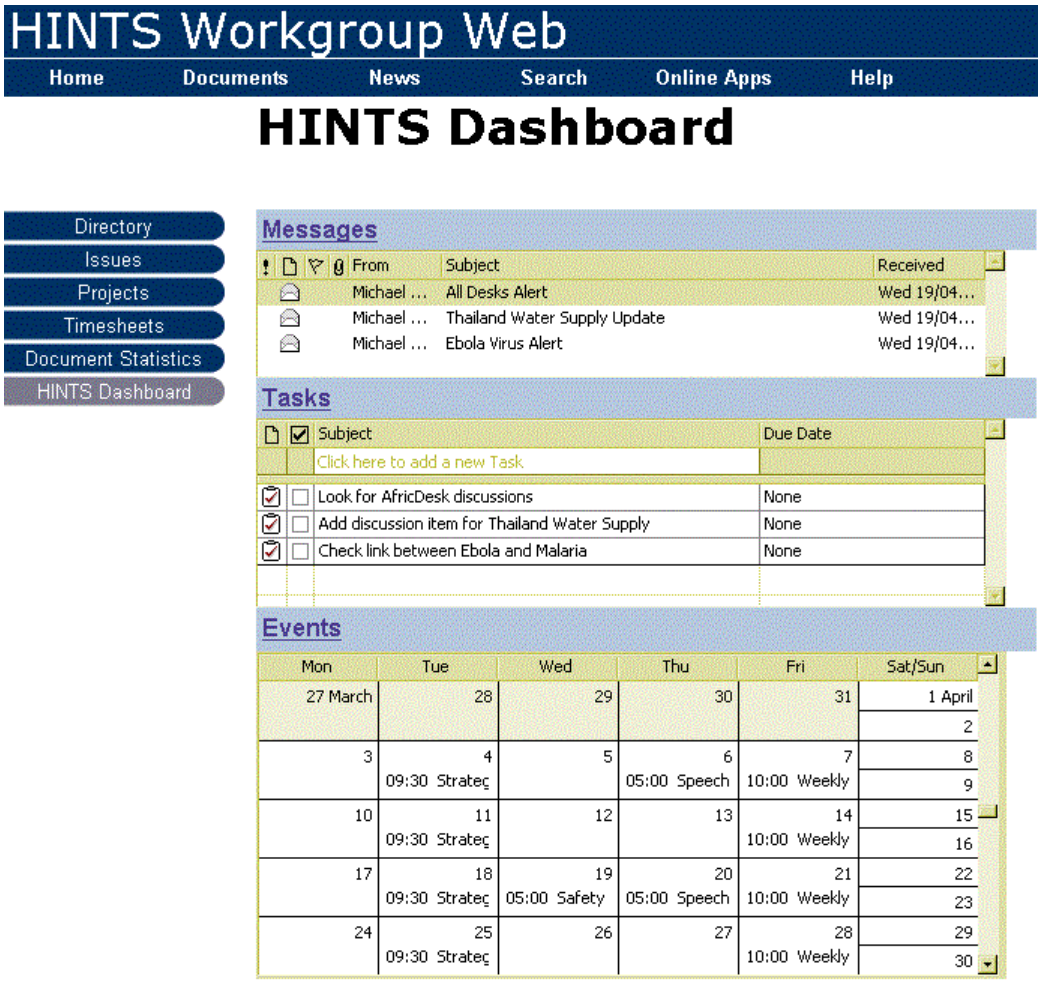

Figure 6. HINTS Digital Dashboard.

Team members view the contents of the dashboard shown in Figure 6 via the web browser. The messages, tasks and calendar views are fully interactive, and draw their information directly from Outlook. However, all information in the web page can be drilled down for more detail, and new messages, tasks and events can be entered without the need to use Outlook. We see this as a convenient way to restrict team members to those aspects of Outlook material to the operation in hand, without interfering with an individual’s own messages, tasks and so on. Additionally, team members need not be trained in the use of Outlook to take full advantage of a digital dashboard.

Of course, the standard online applications that are part of the workgroup web are only part of the processing needed in projects like HINTS. Health intelligence documents need to be processed to generate metadata XML summaries, and these summaries, in turn, must be stored and searched in the document repository. Specialist applications are required to perform these analyses and searches. However, additional web pages can be added to the

workgroup web to trigger the execution of the specialist applications and display the results returned.

Another example of the use of MSMQ events to aid collaboration in the HINTS project appears in Figure 7. Most health intelligence has a geographical dimension, and there is a need to show the location or locations referred to in the document contents. As part of the experiment we constructed a web page with a map image on which are shown document folder images. These folder images are actually ActiveX controls that expose Geospatial coordinates. Simple scripts within the web page move the controls to their appropriate position and update this on a regular basis. Each folder control also listens for MSMQ events, which may trigger a change in position or cause a new control to be created. In a full implementation, clicking on the control will take the user to the appropriate document folder repository.

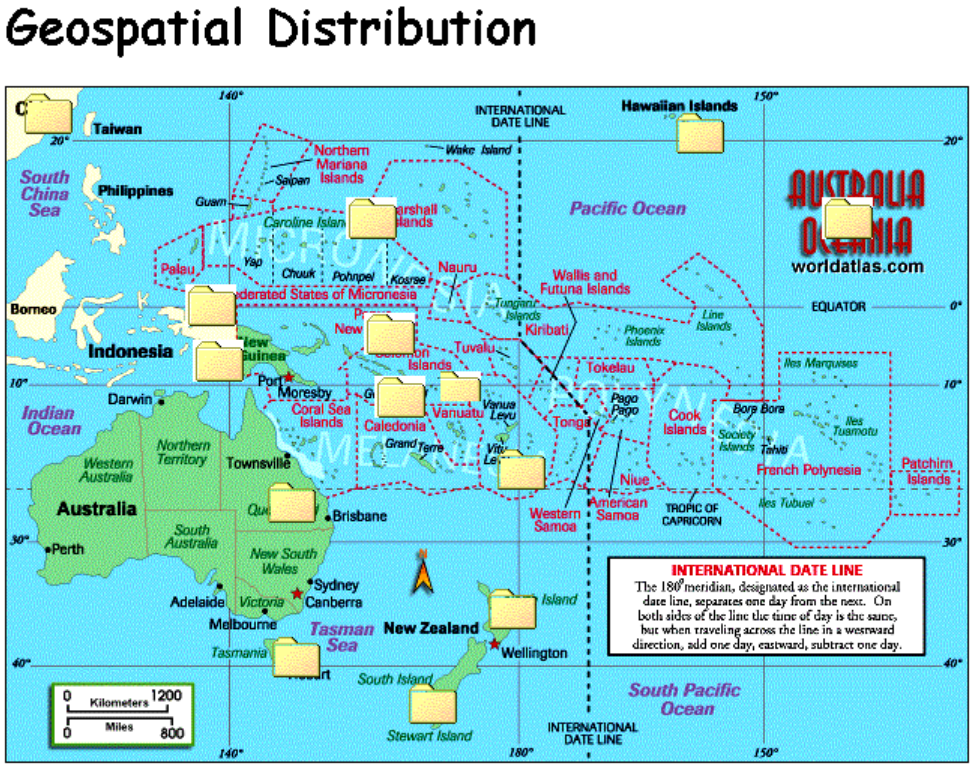

Figure 7. MSMQ Controls for HINTS Geospatial Display.

This experiment shows that with minimal programming a custom interface can be developed in a matter of a few days that is tailored to a particular commander's or team leader's needs. Doing this within the Microsoft environment using Office 2000 tools and leveraging Windows DNA infrastructure (MSMQ, SQL Server, and so on) can provide robust and scalable situation awareness for the commander at marginal cost in terms of time and effort. Obviously, providing real-time, pre-filtered information to the commander in his preferred format frees his staff to better serve him by concentrating on their primary roles.

5 Extending Office 2000 to Support C2 Specific Requirements

As we have shown, Office 2000 offers very powerful collaboration support. However, there will be requirements specific to command and control that it does not directly address. Therefore, we decided to get some hands-on experience to gauge how easy or difficult

extending Office 2000 is by developing some C2 motivated applications. Office 2000 is a component-oriented system based on ActiveX. It is an "open component system" in that it exposes over 800 components to the application developer. There is an "Office 2000 Developer" suite of tools, documentation and examples to help programmers build custom solutions. It is helpful to think of the Office 2000 applications such as Word or Outlook themselves as just examples or instances of applications assembled from the underlying components. Programmers can essentially develop their own customized versions of these applications by using the components. This can be done in languages ranging from C++ and Visual Basic to scripting languages such as JavaScript in HTML.

In this section we describe two Office extensions: a *Discussion Extractor* and an *Annotation Service*. Before we go into detail on these extensions we remark on their development. Two second-year students at the University of Queensland working under a DSTC sponsored summer program developed these applications. We found it remarkable given their limited programming experience they were able to understand and use the many complex tools and techniques required to accomplish this task in an eight-week period. Also, we used all of the tools described in the last sections to collaborate in a distributed manner. We build a work group web for the summer project; this was updated with daily progress reports, new software releases for testing, etc. The discussion feature, web folders and notification service proved very useful. Additionally, we used MSN Messenger and NetMeeting on an hourly basis to answer questions and give guidance.

5.1 Discussion Extractor

A decision support tool developed by DSTO called Group Enabled Meetings System (GEMS) motivated the Discussion Extractor extension. GEMS permits a number of users to conduct real-time "threaded discussions" during a meeting. The goal is to support and record the decision process and help organize the generated artefacts. GEMS is implemented as a large, purpose-built Lotus Notes application. In learning about the new collaboration features of Office 2000, specifically discussions, we thought at least 80% of the functionality of GEMS was covered "out of the box." However, there is no support for manipulating the discussion items after they are entered as in GEMS. Therefore, we chose to build a *Discussion Extractor*. The Discussion Extractor permits the user to query over the discussion items database and format discussion threads into Word documents or PowerPoint presentations.

An example showing how the discussion feature of Office 2000 appears to the user is seen in Figure 8. (This is the home page for the student's summer project web site.) A button on the top menu of IE activates discussions. Clicking this button opens the discussion bar at the bottom of the window. There are two types of discussion items: about the document and "inside" the document. Discussion about the document is shown in the discussion bar at the bottom of the window and discussion items "inside" the document are shown as note icons located at each paragraph. In the figure a discussion item or note is to be inserted at paragraph "Annotation Server Project Presentation".

The Discussion Extractor is implemented as a "COM Add-In." This is the standard way to extend Office applications. A COM Add-In is a wrapper or interface specification that plugs into Office applications and contains application specific code. The COM Add-In appears as a menu item under the Tools menu. The Discussion Extractor operates as follows. When the user selects this option a dialog window appears. From this window the user specifies a query over the discussion database. (Discussions are stored in SQL Server on the host.) The user can

make a query based on documents, discussion topics, users and date ranges. The query will return a set of discussion items. If the user is in Word the items will be formatted into the Word document. If the user is PowerPoint a series of slides will be produced. The Word document and the PowerPoint slides reflect the structure of the discussion. That is, at the top level is a discussion item and each reply to that item is indented underneath it. The author and dates are also shown. This extension permits users to extract, in a formatted presentation, the discussions about any document while running Word or PowerPoint. This extension supports recovery or explanation of such things as the design history of a software project or the rationale for military course of action development.

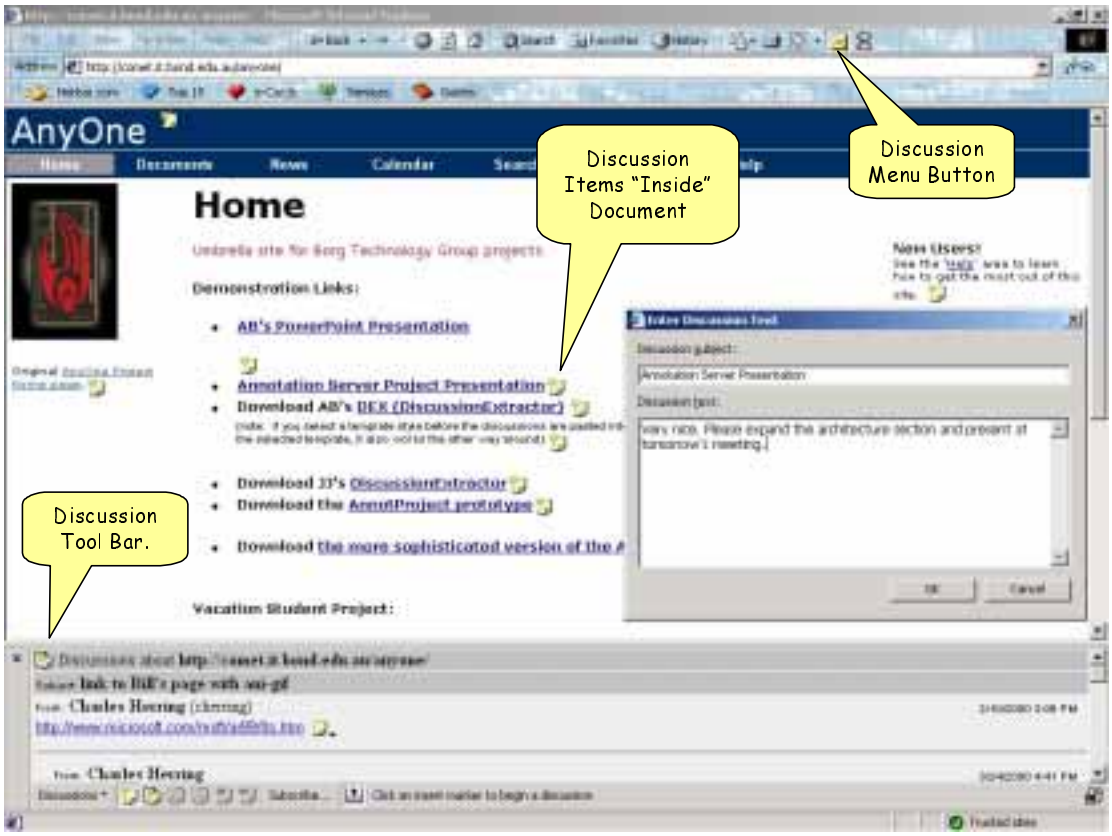

Figure 8. Discussions in Internet Explorer.

5.2 Annotation Service

Another extension we thought useful would be the ability to make graphical annotations to any document viewed in Internet Explorer. That is, the ability to mark-up a document with lines, shapes (rectangles, circles, etc.), text labels and highlighting. This would bring the drawing capability of PowerPoint to Internet Explorer (IE). The user can make arbitrary annotations on any document viewed in IE. Those annotations are stored relative to that document (URL) on the IIS/OSE web site. They are stored in SQL Server (just like discussions) and can be retrieved and displayed by any user with login permission to the web site. The annotations can be selectively displayed as in the Discussion Extractor. Thus different overlays can be made by query. The ability to graphically annotate any document in this manner has obvious applications to C2, for example, the ability to annotate map images.

Developing the Annotation Service required deciding on a software approach to displaying the annotations, linking them to specific locations in a document and storing them. The key to displaying arbitrary graphics is Vector Mark-up Language (VML). VML is Microsoft's graphical mark-up language and implemented using XML. The other key is Dynamic HTML (DHTML). (IE5 natively supports both VML and DHTML.) Using these technologies we implemented the Annotation Service in Visual Basic. The VB application takes user's input (such as moving the mouse to draw a line) and inserts a small JavaScript program into IE. That is, using DHTML's capability to be dynamically modified, a script program is inserted into the document that then displays the graphical annotation. All of the user's annotations are stored inside IE at a known place (inside the document) and when the user choses to save the annotations they are retrieved and sent to the SQL Server database on our host web site. Note that we are not modifying the original document, as it may be located on any web host. We are using the ability to manipulate the Document Object Model (DOM) component that holds the document inside IE once it is downloaded to the users machine. We store the annotations on our web site and they can be retrieved and displayed again when the user revisits the document.

6 Federated Image Management and Dissemination

Imagery and Geospatial information are recognised as a critical foundation data in any C4ISR system and organisations are collecting large amounts of foundation data to meet a wide range of possible situations. In the Defence area the ability to federate multiple distributed data sources is critical to the efficient operation of distributed forces. The Image Management And Dissemination (IMAD) system [Grigg, et al 1999], written in Java/CORBA, has a client-server application structure, allowing geospatial images from a federation of image libraries (databases) to be queried and retrieved by a wide-reach of clients. Image libraries have a local domain, with their availability, unavailability and naming reference fostered through a CORBA trader, local per domain. Different types of clients (from on-field laptops to backoffice workstations) are available for image query and display, differing in compute power and target exploitation of images. Clients request images over geospatial bounding boxes, and all image libraries in the federation are polled to determine if they have overlapping images. The mode of retrieving a particular image is determined from the type of client and its connection bandwidth to the image library server.

The architecture of IMAD is described in Figure 9; each image library server is accessed via a GIAS compliant interface [GIAS, 1998]. Each IMAD component registers its service type, node name and service name. When a local trader receives a request for a service type, it passes the request to currently linked remote traders. It then returns object references of its results as well as those from the remote traders. Each image library comprises image files and a relational database storing image (search) metadata, such as latitudes and longitudes of an image's bounding box.

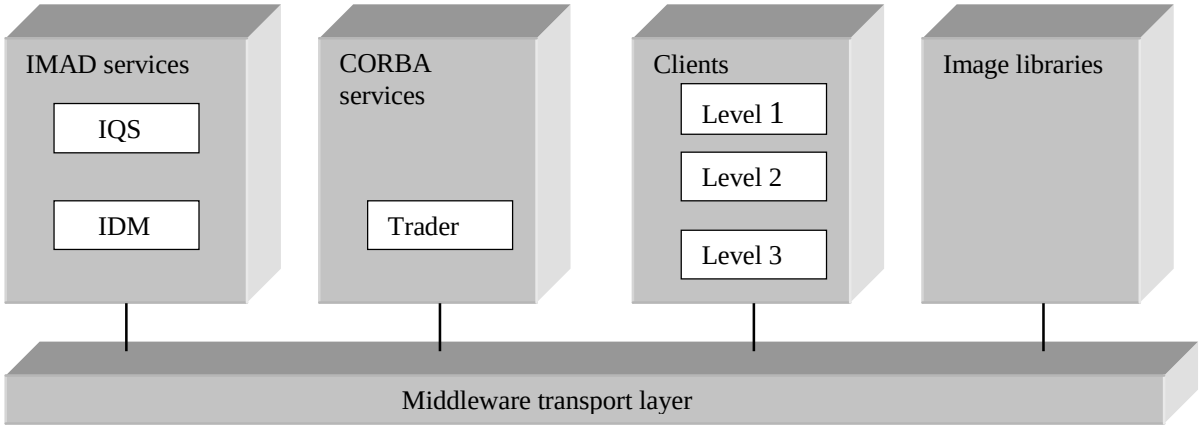

FIGURE 9: The current IMAD architecture

The IMAD process has an image query and an image dissemination part. The image query part determines the image libraries that contain images overlapping the user-specified area of interest. A client first contacts the local trader to obtain an Image Query Manager (IQM), essentially a factory object, for connection to an Image Query Service (IQS). Queries formed from input fields of the client interface are then submitted to the IQS. In order to perform the query, the IQS obtains references to *all* available libraries from the local Trader. The image dissemination part follows user-specified selection of which libraries to target. An image dissemination service (IDS) is determined by a Context Manager, again a factory. The Context Manager appropriates the quality of transmission in an image dissemination service generation, using the likely network bandwidth between the client and the library server, its level, the size of the image (pixels) and the size of the region of interest. Currently FTP and socket-based image retrieval services can be generated. CORBA based transfer is not considered feasible currently.

Our proposal for the IMAD architecture under DNA is depicted in Figure 10.

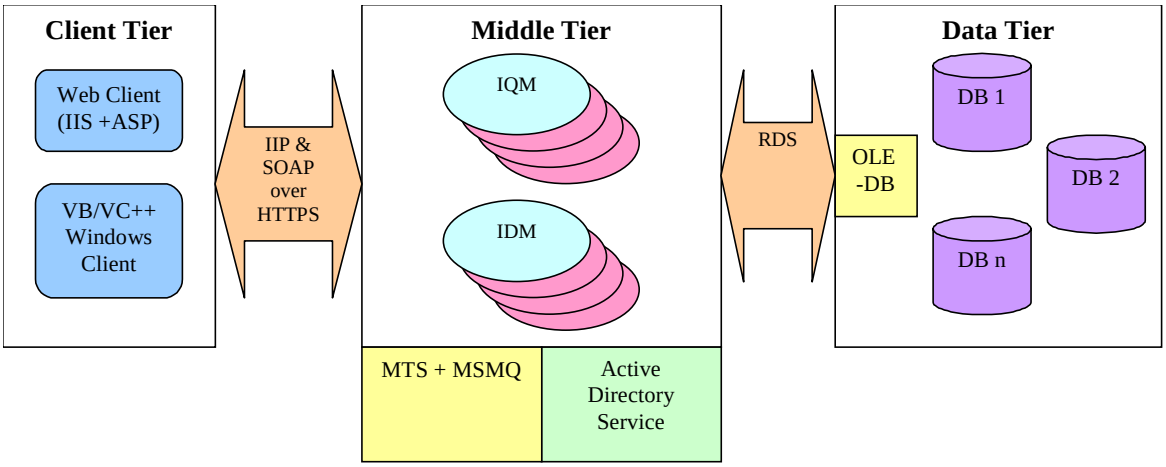

Figure 10: IMAD architecture under Microsoft DNA

6.1 Client Tier

Clients within the IMAD architecture can be implemented using a variety of approaches. *Level 1 clients* can be implemented using a combination of: Internet Information Server (IIS), Active Server Pages (ASP) and ActiveX components (both in the Web page and embedded in server side ASP scripts.). *Level 2 clients* can be implemented as stand alone Windows applications using either Visual Basic (VB) or Visual C++ (VC++) and ActiveX components. *Level 3 clients* can be supported by implementing specialised tools to export data in the appropriate format using either VB or VC++ and ActiveX components. In order to facilitate reuse, a core set of ActiveX controls that can be shared between two or more types of client. In particular, we bundle the IMAD Query Manager, as an Active X control, into the clients, for improved efficiency over the current IMAD design. We advocate a broader range of clients interacting with IMAD through the use of XML for client and middle tier communication. Under the recently emerging SOAP standard, XML can be transported through firewalls over secure HTTP. In place of a CORBA trader, the dynamic location and discovery of the appropriate middle tier services occurs through the Active Directory (AD) service (described in more detail below).

6.2 Middle Tier

Microsoft Transaction Server (MTS) controls access to business logic components, in this case the IQM and IDM components. In doing so, MTS provides object brokerage, object instantiation from components, security checks to prevent unauthorized access to components; two-phase commit transactional management; resource management of system resources, memory and threads; and load balancing which will be provided in future versions of MTS.

To ensure that requests to a data source wrapper do not exceed the capacity of the backend data sources, a limit can set on the number of simultaneous connections (for example limited to the number of connections in the connection pool). Additional requests should be stored in MSMQ. Also, requests in the message queue could be prioritized, e.g. users with more important requests (e.g. senior ranking officers) having their data demands met first. Finally, the message queue can enable work asynchronicity to be built into the system. For example, a mobile user may submit a query for processing, disconnect from the network, and reconnect later to fetch the results.

A major optimisation results from the exposure of image library meta-data through the Active Directory service. A single AD service is proposed for the whole IMAD federation, which could easily evolve into a hierarchy of ADs for improved performance and resilience, as the number of image libraries significantly increase. This presents a central source of image library searching, undertaken by the IQS, which currently does a broadcast mode of searches to *all* image libraries. To enable this, all image library servers would be registered as objects within an AD schema. Along with the domain name and IP address attributes, all exposed meta-data for the image library would also be stored in the object's type definition. Thus, an image object (with the meta-data as attributes) would be associated with an image library object that in turn would be associated with an image library server object.

Note, the distinction made between image library server and image library objects. This provides a number of flexibilities. For example, a number of image libraries (databases) can share the same image library server under this scheme. Also, the image library server and image libraries can be manipulated as separate resources. For instance, an image library can be blocked from access while the server can still continue to be accessed - for asynchronous query (message) forwarding to the server.

Under this configuration the IQS and IDM processes would now work as follows. All image objects in the AD would be searched to satisfy the user query. Under an object scheme (as opposed to a relational scheme), this would involve a search of all image objects – which would be fast if the AD was cached and the appropriate attributes indexed. All satisfying image objects would be stored in the hit-list together with the image library and server references, which would be displayed for levels 1-3 clients. In the IDM phase, the IP addresses of the servers and image library names would be used for query "connection" purposes. These would be obtained by reinterrogating the AD.

A central directory for image searching means that image updates in a library would have to cascade to the central directory. Libraries, under strictly autonomy, however, should be free to do so "as they please". Thus, some versioning mechanism could be introduced into the Active Directory to reflect the currency of the updates on an image as well as a image library basis. This would be used to either flag the IQS or the user for local image library searches, to ensure currency of the result sets. For instance, the library might expose the fact it has undergone major updates, which are not reflected in the directory. Either the IQS would then expand its search to the image library or it would notify the user (as a status field in the result-set), leaving it up to the user to request the expanded search.

Note the qualification of *exposed* meta-data. This allows libraries to choose to their own autonomous library structures and expose the relevant meta-data accordingly. Thus, the Active Directory would require a flexible, dynamic schema mechanism that allows for different image object types to be associated with the image libraries.

If more complex queries were required across heterogeneous libraries another approach would be to use the distributed query capability of SQL Server to provide a federated query service. Both the integrating databases and the libraries would be located using the Active Directory. In both cases the interface to the underlying data would be via OLE DB as described in the next section.

6.3 Data tier

6.3.1 OLE DB

OLE DB provides more generalised access to data sources than ODBC/JDBC because it includes relational and non-relational sources. Providers of the sources are required to map between OLE DB data structures and their local data structures. Suitability of OLE DB access for relational sources, given the inclusion of non-relational cursor-like query navigation, remains uncertain. So too does the efficiency of driver bindings with respect to the database interface manager – i.e. does the driver run in the same process space as the manager or does it run locally in the database server? This will determine the efficiency of OLE DB calls.

We recommend that OLE DB be run as its own separate process and accessed via the Remote Data Source (RDS) COM interface. The main reason for doing this is to enable connections to be maintained by the connection pool even after objects instantiated by the IQM or IDM have completed their task. A second reason is that it will mean less overhead as each instance of the IQM and IDM will not need to have their own copy of OLE DB. RDS COM interfaces can be advertised in the Active Directory Service and dynamically located at run time by the IQM or IDM.

6.3.2 SQL Server

In the current IMAD architecture the image data is held on a file system. One possibility is to fully exploit SQL Server's storage and access for both meta-data and images files. The main reason for doing this is simplified management. SQL Server offers a suite of administration and management tools that can make tasks such as backup, replication, linking and federation of data sources much simpler. In particular, the distribution, data export and replication facilities provided in SQL Server allow larger storage configurations of image libraries, allowing clustering of libraries where a set of libraries share a single schema but have separate replicates (like a distributed database) or where separate image libraries have the same physical storage (for management economy). A motivation for clustering is the provision of particular IMAD sites for specific purposes and missions. Reliance on SQL Server functionality for such situations provides a single point of control for replication/distribution where updates are automatically cascaded.

Another benefit is that specialised image content search functions can be seamlessly integrated with standard database queries. Under a separated storage strategy, meta-data (database) and image (file) searching will be separate and lower level programming will be required for the integration of "result sets". A database querying language allows pre-written as well as custom-supplied functions to be specified as part of the query, thereby accommodating a large and typically unforseen query requirements. It should be noted that full query optimisation would not be possible under this approach. Moreover, the querying of database image objects is likely to be slower than in direct image file access because certain aspects image querying will be controlled by DBMS's data manager, e.g. caching some parts of the image object.

The main problem of including images into a database is the loss of control at the file system level to fully exploit large-scale active (hierarchical) storage devices. Such file managers have their own schema, and it is not clear how image files can be mapped in a one-to-one fashion into elements of these schemas. DBMSs typically allow larger grains of storage areas like file systems and partitions to be allocated to individual tables/indicies or tablespaces/indexspaces. This mapping does not easily lend itself to mapping into more particular file storage schema, which provides clustering between image sets, images, tiles and various granularities thereof, and physical elements of storage devices. Clearly, a dedicated provisioning is required by a DBMS, which in turn requires a joint collaboration by database vendors and large-scale storage device vendors. This remains an open issue [VLDB 1999].

Added to this is the more open issue of image content querying. The same levels of content search cannot be guaranteed for images as with standard database records. Content in an image is subject to uncertainty depending, among other things, on the quality of image and the structural recognition capacity of contained and targeted objects. As apparent through

current research efforts [Nepal et al, 1999], the incorporation of uncertainty in image searching and the provision of general-purpose image query languages, still remains open.

6.3.3 Image data servers

In image information systems both the size of individual images and the number of images in a collection may be very large. To manage this volume of information a number of commercial and experimental systems are being developed to store, manage and disseminate image information. For level 3 clients it is assumed an FTP mechanism is used to transfer data to the client. For other client types one will require a more dynamic mechanism that delivers image data on an as required basis is needed. Some of these systems, described in the following, support delivery of imagery in a multiresolution on demand basis:

Flashpix and IIP: The Digital Imaging Group [DIG, 2000] has developed the Flashpix Image format and Internet Imaging Protocol (IIP) for distribution of high-resolution imagery in Internet like environments. This format and protocol support tiled multi-resolution imagery and access to data only as required. Imagery servers using IIP are available commercially for distributing high-resolution imagery. Several commercial imagery products have built in support for Flashpix. The Flashpix format stores imagery in multiple resolution levels to allow quick access to the image at multiple resolutions over network connections. Within each resolution level the image is tiled to allow fast access to arbitrary regions within the image. Each tile may be stored in uncompressed or JPEG compressed form. The IIP specification supports socket and HTTP connections. The protocol supports requests for regions and resolution levels within the image data. Some processing operations such as affine transformation and contrast enhancement can optionally be supported by the protocol. This protocol allows a URL based request to specify the image to display together with the resolution and region required. Web pages (or other clients) can use this protocol to construct dynamic pages that allow the user to pan and zoom through very large images without excessive bandwidth requirements. An IIP server has been used in this project to allow display of very large images. ActiveX components are available from third party suppliers such as HP's OpenPix.

MrSID: MrSID is a commercially available wavelet based image format that is available from a single vendor [MrSID, 2000]. The use of wavelets avoids the need to store multiple resolution levels and tile the image. This format allows the viewing of very large raster datasets on standard hardware. Viewers and plug-ins for Internet browsers are available and several third party vendors support the format. To create MrSID imagery a commercial product from the developer of this format must be purchased. An image server, browser plug-ins, and JAVA applets are available for using MrSID imagery in Internet environments.

ERMapper Image Server: The ERMapper Company has also developed a wavelet based compression method and web based delivery protocol. Plug-ins for web browsers and several commercial products are freely available. An image server is available to deliver imagery compressed in their format across the Web on an on demand basis.

6.4 Value-added IMAD functions

In addition to the optimisations and improvements designed into the new IMAD system, we propose the following value-added functionality, which now becomes possible.

6.4.1 Dedicated sites

The idea of dedicated sites is to provide sub-IMAD systems for particular operational missions. As with infrastructure, logistical and personnel deployment for missions like East Timor, applications and databases are often pre-fabricated out of existing resources as supporting services. Under this business practice, subsets of image libraries and even search results (using Web folders) could be cached into a cluster of image databases which are updated as their master image libraries are updated – within the *scope* of the sites (e.g. only image within a geospatial boundary or containing particular geospatial features). For such a configuration, SQL Server master-slave replication strategy could be set up, using a predicate to specify the replication, i.e. site, scope.

A further change to IMAD would be to allow for the specification of a site scope. This means that the IQS and IDS would only operate on image libraries in that scope. Such a scope could easily be incorporated into the IMAD Active Directory Schema as a separate objects associated with image library objects or as one of their attributes.

6.4.2 Automated, dynamic meta-data generation

In the core model architecture, image searching takes place through a global Active Directory with local schemas at image libraries allowing for more refined searches as well as image dissemination. There is no constraint that these schemas be static, i.e. predefined. Indeed this would be undesirable for the global schema, which has to incorporate a variety of meta-data exposures from image libraries. A partially dynamic strategy is to allow exposures at fixed times, e.g. as part of the nightly maintenance run of the Active Directory server. Certainly, this strategy follows the static nature of image meta-data generation (which is mainly manual).

A fully dynamic strategy would be to allow meta-data to be generated “on the fly” and exposed at any time to the Active Directory global schema. This solution would become more compelling as image content searches become more feasible, where image scans could be undertaken for pre-conceived searches, and then both local and global schemas would be updated (in two phase commit). From a functional point of view, interfaces at both the image library and Active Directory levels would allow updates to their schemas - the caller requiring the update simply hands over an image update script through the interface. Performance and security issues however loom. Schema updates should at worst block searches to images of the updating image library object. A versioning strategy for schema updates would ensure such efficiency. A security role model would be require to authorise which users could undertake schema changes.

7 Non-Functional Issues

The discussion and experiments described in this paper have focused on providing functionality that would be useful in C2 environments. There are also non-functional issues that must be addressed for developing C2 systems. While these have not been addressed in detail in this work we can comment on some of them as follows.

When using COTS software and Microsoft software in particular the issue of security is often raised. While it is true that a popular operating system and software will be subject to attack

due to its widespread use this does not mean that abandoning these products is the best solution. In fact this very popularity has produced a wide market for security related products that can be used to secure these systems. All operating systems have had security related issues in the past [Denning, 1990], [Tanenbaum, 1997] so that the only way to produce a secure system is to put procedures in place to understand and account for any limitations in the computer system. Defence departments have accreditation procedures for operating systems and the NT operating system has been accredited for use on secure systems. The Windows 2000 system has increased security related features and will also undergo such accreditation.

Another issue in some C2 systems is the availability of bandwidth for the applications. Between fixed sites with good communications the use of commercial software is not a problem but over low bandwidth tactical links the use of software that assumes Ethernet like network connectivity can cause problems. While the use of the previous experiments in low bandwidth environments is an area for future research, the following general comments can be made.

The experiments that delivered capability via a web page should work over any link that can support web connectivity. Some of the collaborative features require particular port access in a TCP/IP environment. The use of MSMQ as the connecting protocol to a backend service in the digital dashboard example illustrates how a protocol that can tolerate low bandwidth and temporarily disconnected environments would assist in deploying these applications to low bandwidth environments. Due to the increasing commercial interest in mobile and disconnected computing the ability of commercial tools to handle these situations is increasing. For example Windows 2000 includes the ability to work on file systems and web pages offline with the results synchronised on reconnection.

8 Conclusions and Future Work

In this paper we have described the software problems inherent in the current approach to developing C2 systems. We have shown how most of these can be eliminated and a new generation of C2 systems developed based on Microsoft software and architecture. This approach has the advantage of being able to rapidly take advantage of new COTS technologies in addition to the reduced need for specific coding development. The use of Windows 2000 DNA as our C4ISR Architecture enables multiple technologies to be rapidly engineered and integrated with minimal effort over a short time frame.

Based on the experiments described above we believe that component-based, Microsoft COTS products, such as the 60 Minute Intranet Kit, the Digital Dashboard and Office 2000, show the way forward for future C2 systems. To reach the level where end users can “generate a C2 system in an hour”, as the title of this paper claims is possible, a first step is to realize a number of C2 Wizards. We foresee a set of C2 Wizards for generating workgroup collaboration webs for each staff function and at each level of command. In order to accomplish this, a methodology and support tools are required to aid in development of the wizards. These tools are very lightweight however and reuse most of the same underlying components and content. Our approach places the emphasis on C2 domain analysis and away from software development. A goal is that the end users will learn to build their own wizards. As the initial set of wizard-generated applications emerge, more information-integrating applications such as the Commander’s Digital Dashboard can be assembled. Once again, a

wizard-based approach can be taken to permit commanders to generate their own view into the information space. We believe a small set of "C2 nugget" components can go a long way toward providing the commander with a flexible situation awareness environment.

Microsoft has been pursuing an Internet-based strategy since 1995. That is, the incorporation of Internet technologies, such as TCP/IP, HTML and now XML, at a fundamental level throughout its products. The Microsoft products chosen for our experiments illustrate this. We believe the trend will continue and result in a high level of standards-based consistency and interoperability. Now, DOD developed the basic Internet technologies specifically for C2. In turn, the Internet has had a profound affect on military thinking leading to the current "Network Centric Warfare" philosophy [NCW, 1999]. The experiments shown here and the approach advocated for future C2 development is the most direct route to achieving the goals of Network Centric Warfare.

We intend to pursue the Microsoft approach in several dimensions. Firstly other technology experiments in specific C2 application areas will be conducted to increase the demonstrated depth of the approach. Secondly several of these experiments can be combined to illustrate the provision of the breadth of functionality typically required in C2 systems. The concepts of C2 wizards will be developed to enable the rapid construction of various components into a customised C2 system for end users. To compare the effectiveness of systems constructed in different technology environments a selected system, such as the IMAD system could be constructed in multiple technology domains and then compared on the basis of an agreed set of performance and effectiveness metrics.

9 Acknowledgement

The work reported in this paper has been funded in part by the Co-operative Research Centre for Enterprise Distributed Systems Technology (DSTC) through the Federal Government's AusIndustry CRC Program (Department of Industry, Science & Resources), and through the Department of Defence EXC3ITE project.

10 References

[Denning, 1990] Denning, Peter: "Computers under attack: Intruders, Worms, and Viruses", Addison-Wesley, 1990.

[DIG, 2000] The Digital Imaging Group, http://digitalimaging.com.

[DII, 2000] Defence Information Infrastructure Common Operating Environment Home Page, http://spider.dii.osfl.disa.mil/dii/.

[GAO, 1998] U.S General Accounting Office, "Defense Information Superiority - Progress Made, but Significant Challenges Remain.", NSIAD/AIMD-98-257. 08/31/98 (Letter Report), http://www.access.gpo.gov/cgi-bin/getdoc.cgi?dbname=gao&docid=f:n198257.txt.pdf

[GIAS, 1998] *Geospatial and Imagery Access (GIAS) Specification 1998*, US National Imagery and Mapping Agency, http://www.nima.mil/aig/

[Grigg, et al 1999] Grigg, M.; Whitbread, P.; Irving, C.; Lui, A.; and Jana, *Component Based Architecture for a Distributed Library System*, Distributed Multimedia Systems Conference 99, Aizu, Japan, July 1999.

[MrSID, 2000] LizardTech Website, 1999, http://www.lizardtech.com/ .

[MS-DD] Microsoft Digital Dashboard Kit, http://www.microsoft.com/solutions/km/DigitalDashboard.htm

[MS-DNA] Windows DNA, http://www.microsoft.com/dna/

[MS-KIT] Microsoft, 60 Minute Intranet Kit, http://www.microsoft.com/office/WebTour.htm

[MS-O2K] Microsoft Office 2000, http://www.microsoft.com/office

[MS-OSE] Microsoft Office Server Extensions, http://www.microsoft.com/office/deployment/default.htm

[NCW, 1999] Alberts, D; Garstka, J.; and Stein, *Network Centric Warfare*, C4ISR Cooperative Research Program, www.dodccrp.org, 1999.

[Nepal, et al, 1999] S. Nepal, M.V. Ramakrishna, J. Thom; "A Fuzzy Query Object Language (FOQL) for Image Databases, in Proceedings of the Sixth International Conference on Database Systems for Advanced Applications, April 1999, Hsinchu, Taiwan.

[SPROTT, 2000] David Sprott; "Componentizing the Enterprise: Application Packages", Comm. of the ACM, Vol. 43, No. 4, pg 63-69, April 2000.

[Tanenbaum, 1997] Tanenbaum, A; and Woodhull, A; "Operating Systems: Design and Implementation", Prentice-Hall, 1997.

[VLDB 1999] Industry session at VLDB'99 – see Proceedings of Very Large Scale Databases 1999, Edinburgh, September 1999.