Web-Based Information Management of Maintenance Errors in Aviation Mishaps

George Zolla, Pat Flanders, Tony Boex
Naval Postgraduate School
Monterey, CA 93943
gazolla@nps.navy.mil

Abstract

This paper describes a safety information management system designed to capture maintenance factors that contribute to aircraft mishaps. The Human Factors Analysis and Classification System-Maintenance Extension taxonomy (HFACS-ME), an effective framework for classifying and analyzing the presence of maintenance errors that lead to mishaps, incidents, and personal injuries, is the theoretical foundation. An existing desktop mishap application is updated, a prototype web-based model is developed, data-mining techniques are discussed and an Asynchronous Distributed Learning (ADL) module is conceptualized. These tools facilitate data collection, organization, query, analysis, and the reporting of maintenance errors that contribute to aviation mishaps. Together they represent a complete, robust system for analyzing aircraft maintenance mishap related factors anywhere at anytime.

1. Introduction

From 1950 to 2000 Naval Aviation had great success in substantially reducing its Flight Mishap (FM) rate (Figure 1). However, the proportion of mishaps attributed to human error remained at a relative constant rate of about 80 percent [8]. In 1996, a Navy F-14 Tomcat crashed shortly after taking off from Nashville, Tennessee killing both aircrew and three civilians on the ground. Since the cause of this mishap was exclusively human error, Department of the Navy (DON) leaders established a Human Factors Quality Management Board (HFQMB) to significantly reduce mishaps caused by human error by identifying systemic improvements in the processes and systems that guard against error and enhance human performance. The HFQMB's goals were to reduce the Naval Aviation Class A (most serious) Flight Mishap (FM) rate by 50 percent by FY 2000 and 75 percent by FY 2006 [6].

The HFQMB's initial thrust was to conduct an extensive mishap data analyses focused on human factors. The Naval Safety Center (NSC) developed the Human Factors Analysis and Classification System (HFACS) taxonomy to capture aircrew errors in Naval Aviation mishaps. This taxonomy identifies areas for potential intervention by describing factors that may be precursors to accidents. The resulting HFACS taxonomy focused solely on aircrew errors and identified both active failures and latent conditions within four categories: 1) unsafe acts, 2) pre-conditions for unsafe acts, 3) unsafe supervision, and 4) organizational influences [2]. NSC has adopted HFACS for analyzing human error in Naval Aviation mishaps and targeting appropriate prevention [2].

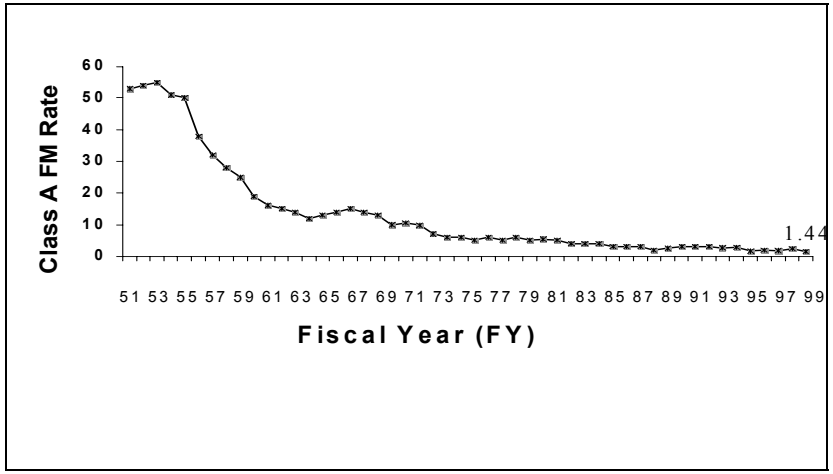

Figure 1: Naval Aviation Class A Flight Mishap Rates for FY 1950-1999

(NCS, 2001)

After the establishment of HFACS in FY 1999, Naval Aviation enjoyed its lowest mishap rate ever. Even with the reduction in the mishap rate, the HFQMB's goal to reduce human error related mishaps by 50 percent by FY 2000 was not achieved [7]. During this period, a study discovered that HFACS could be extended to cover maintenance errors [10]. Hence, a Maintenance Extension (ME) for the HFACS taxonomy was adapted to classify causal factors that contributed to maintenance mishaps [9].

The HFACS-ME classification system contains four first order, ten second order and thirty-four third order human error categories. The first order categories consist of: 1) Management Conditions, 2) Working Conditions, 3) Maintainer Conditions, and 4) Maintainer Acts. A mathematical analysis of 470 Naval Aviation Mishaps by Schmorrow [11] determined the HFACS-ME taxonomy was an effective classification system for determining trends in aviation mishaps.

In the next section, we will discuss a desktop database model of HFACS-ME. A web-based version for Internet and intranet use will then be introduced. Possible future enhancements will then be outlined followed by the conclusions.

2. MEIMS

Building on Schmorrow's research, Fry [5] developed the Maintenance Error Information Management System (MEIMS), a desktop database application based on the Microsoft Access 97 Database Management System for the analysis of maintenance related mishaps. MEIMS used additional mishap data which led to a refinement of HFACS-ME, making it more comprehensive and accessible.

Fry's rudimentary MIEMS tool was further refined by Wood [12] and developed into a working prototype, for U.S. Navy Fleet testing and evaluation. A usability study of the prototype MEIMS tool determined that it could be an effective system, not only in determining trends but providing information for mishap prevention efforts. Wood's study identified the need for MEIMS to incorporate improved HFACS-ME definitions, improve the user interface, simplify data entry procedures, and include a narrative of common mishap scenarios.

A grant from the Federal Aviation Administration (FAA) and support from civilian air carriers expanded the original scope of the project to include the design and implementation of a civilian variant of MEIMS. The FAA also directed that an upgraded desktop database model of MEIMS be created that would be compatible with Access 2000. Ideally, the updated database model would [4]:

1. Capitalize on object-oriented technology to provide better performance, greatly increased scalability, and broader opportunity for code re-use.
2. Provide multi-user access to the database in a true client-server environment while maintaining the ability to function as a small footprint, easy to use, standalone desktop application.
3. Provide a full-featured interface for web-based access.
4. Isolate itself, as best as possible, from compatibility changes in new versions of software. This includes development language, database engine, and operating system.

The migration of MEIMS from *Access 97* to *Access 2000* proved to be difficult because of the differences in the built in programming language *Visual Basic for Applications* (VBA). *Microsoft Access* has built-in functionality to create desktop applications with forms, reports, and embedded support for *Visual Basic for Applications* (VBA). The data in an *Access* database can be manipulated using several different programming languages, Active Server Pages (ASPs) via the web, and via third party add-in tools. A key feature of *Access* over other databases and development tools is its ease of use - it is a very effective rapid application development (RAD) platform. *Access* can be (depending upon the implementation options) much simpler to use for building standalone, small-scale database solutions than many of its competitors. Our development effort focused on adapting the previous *Access* "structured" systems to *Microsoft*-based object oriented architectures ensuring future scalability and increased potential for code-reuse.

A new feature of *Access 2000* that made it appealing for the HFACS-ME project is the ability to use more than one type of database engine. A database engine is the part of a database management system (DBMS) that actually stores and retrieves data. *Access 2000* provides support for both the *Microsoft JET* database engine and *the Microsoft SQL Server* engine. This is a key distinction. *Access* originally had only one choice as a database engine: *JET*. The main problem with *JET* is that it is not a client/server capable engine. It really performs as a file server. This means that anytime a client wants to request something from a *JET* database, a lot more has to be done on the client-side. The result is a lot of network traffic and unacceptable response times for more than only a handful of simultaneous users. With the release of *Office 2000*, however, *Microsoft* provided a royalty free version of the *SQL Server* engine for use with *Access*. Most significant, (other than it being free) is that it is capable of running on a desktop computer. This change allowed an *Access* solution that had the ability to operate as a stand-alone application using the same engine as the full version of *SQL Server*. Upgrading from a desktop application to a server-based application was no longer an issue because the engine is the same.

One confusing aspect of the standalone engine is the difference in naming conventions between various versions of *SQL Server*. The *SQL Server 6.5 and 7.0* compliant version is called the *Microsoft Data Engine* (MSDE),

while the *SQL Server 2000* compatible variant is called *Microsoft SQL Server Desktop Edition.* Both versions provide *SQL Server* database functionality, but there are significant differences. In order to provide greater emphasis on the distinction between full *SQL Server* and the two Desktop editions, we will refer to both the *SQL 6.5/7.0* and *2000* versions of the desktop engine as *MSDE* unless otherwise stated.

There are automated migration tools designed to port *JET* databases over to *SQL server,* but they are useful only for very simple databases. The *Microsoft Access* "Upsizing wizard" was used on MEIMS with very poor results. Structured Query Language written in *Access* using *JET* did not transfer correctly. Functions written in VBA did not transfer correctly. In addition, the data types used by *JET* are different from those in *SQL Server* and they did not transfer properly. Finally, *Access* uses "queries" in place of stored procedures and queries did not transfer at all. To put it simply, the *JET* database engine is not scaleable and was ruled out as a viable option for the new version of MEIMS very early in the requirements analysis process.

MSDE has a lot to offer in terms of use with *Access* and *Visual Basic*. For example, the desktop engine supports record-level locking, transaction logs, operating-system integrated security under Windows 2000, and many other advanced features of full *SQL Server* (like replication) -- all from *Visual Basic* and VBA. In fact, we found that the *SQL Server* engine actually had an excellent array of options, most formidable of which was the selection of programming interface to access the data in it.

As mentioned earlier, *Access* has embedded support for *Visual Basic for Applications*. The SQL engine, however, is accessible via any language capable of creating COM objects. Using *Visual Basic* or C++, *ActiveX* object-oriented components can encapsulate much of the code that would normally be written within *Access*. These compiled components can reside outside of *Access* theoretically making them less susceptible to version changes and maximizing potential for code reuse. *Access* would just be a client shell and all business logic would be placed in these external components. The beauty of this approach is that the RAD methods of *Access* used to create forms, reports, and controls were still available. In addition, this approach is in keeping with the migration path of a small-scale application to a larger enterprise level one using OLE DB and DCOM. The location of the external components (either client-side or server-side) would define the architecture of the system (3-tier or n-tier). Removing the business logic from *Access* allows MEIMS to grow by enabling modification of component code without making changes (or many changes) to the client code in *Access* or the database elements in *SQL Server*. Since code in the components is compiled, changes in versions of the programming language used to create them are much less significant over the lifespan of the program. This is especially significant for MEIMS because of *Microsoft's* upcoming release of new technologies like *C#* and *Visual Basic .NET*.

Based on the decision to implement components, the next decision involved selection of a COM compatible programming language. In keeping with the requirement for a Microsoft based solution the choices were either *Visual Basic 6.0* (VB) or *Visual C++*. Both C++ and VB are capable of implementing the four data access technologies that we knew we would need. Since *Access* provides inherent support for VBA and *Visual Basic 6.0* is a superset of this technology, VB could provide a single language for use in both *Access* and the components. C++, on the other hand, offered greater support in terms of threading (which is a serious deficiency in VB). A major disadvantage of C++, however, was its added complexity in a program designed for RAD. In the end, the idea of a using VB in all coding for HFACS was truly the key factor in weighing advantages and disadvantages. The final choice for programming language was *Microsoft Visual Basic 6.0*.

Results - The requirements analysis effort coupled with research of data access technologies, programming languages, and trends in *Microsoft* products resulted in the development of a vision for a new MEIMS system. The conceptual framework inferred the following:

1. MEIMS should consist of a *Microsoft Access* client application using external compiled components to encapsulate business processes wherever possible. This would provide greater opportunity for code reuse and mitigate the effects of version changes in *Access*.
2. MEIMS would implement the *SQL Server* database engine and therefore should be developed to connect to an instance of MSDE as well as true SQL server. In order to facilitate differences in these connections, a component would be needed to perform management functions such as installation of the programs, installation of the database, logon options, and starting and stopping the server. Management functions of this depth should be performed using SQLDMO and are specific to each client, therefore, this component must also reside on the client.
3. MEIMS would be a true Client/Server application.

To facilitate these findings, the MEIMS development effort was divided into two phases. Phase I focused on development and implementation of the MEIMS Connection component. Phase II did the same for the MEIMS business logic component. The development of the connection component was executed

first because it involved creating the foundation and environment for the business logic component to operate in. In addition to creation of the connection component and the inherent connection functions, Phase I involved creating the installation programs needed to deploy and configure all the pieces of this operational environment on a wide array of platforms supporting various editions of *SQL Server* and *Windows* operating system. The high-level conceptual architecture is illustrated below.

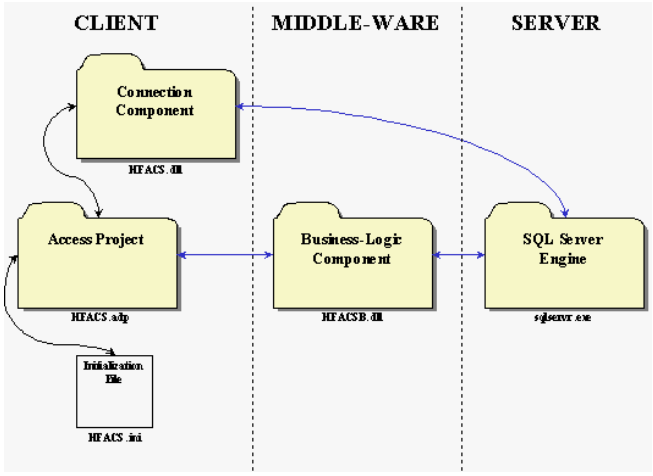

Figure 2: Conceptual Architecture

Within phases, the Spiral Development Model (SDM) techniques were used. The SDM provides built-in methods for mitigating these risks through its use of development stages. Each stage was a normal development project producing a superset of the prior stage and yet a subset of the final system. Planning for each successive stage was structured to exploit the experiences of the former stages and to reduce perceived risk factors in the current and future iterations.

In the end, the methods that described above met all system requirements, greatly enhancing current desktop MEIMS capabilities and providing the means to weather further changes in requirements and application platforms. During startup, MEIMS allows the user to select the database that will be used with the user interface. Selection of the database stored on the local computer allows the entire application to be self-contained. Selection of an enterprise-level SQL Server DBMS allows an organization to securely manage mishap information from a true SQL Server DBMS and allows a system administrator to update records and stored procedures from one central location. Using the Navy and Marine Corps Intranet or a civilian air carrier intranet allows the desktop version to be linked to a database that is behind a firewall. This keeps the mishap data current and confidential.

USER INTERFACE - MEIMS is designed to allow the user to access the database via three functional tools: (a) extracting data using user-created and predefined *queries*, (b) obtaining output from the database via *written reports* and *graphical displays*, and (c) inputting new data via forms. Each function is displayed on separate pages with controls providing user interaction. The system facilitates simple searches for novice uses and complex data-mining operations for more sophisticated users. The following paragraphs provide a description of the user interface.

The Main Menu allows the user to select one of five different options: (a) Query Menu, (b) Graph Menu, (c) Report Menu, (d) Mishap Investigation, (e) Administration and (e) Exit.

The Query Menu (Figure 3) provides the user with the ability to search by single field queries or multiple field queries. The single queries offer one of eight command buttons to extract data by one or more of its fields: aircraft model (F-14, H-46, etc.), aircraft type (tactical aircraft (TACAIR), helicopters, heavy aircraft, trainers, and others), branch of service of the aircraft (USN, USMC) or carrier name for the civilian variant, location of the mishap (ashore, embarked, and detached), mishap classification (A, B, or C), mishap type (Flight Mishap (FM), Flight-Related Mishap (FRM), or Aircraft-Ground Mishap (AGM)), and calendar year of the mishap (1989-1999).

When a single category control button is selected, a sub-menu appears allowing the user to define the exact description of the category via a combo box. Upon selecting the "View" control button, a Maintenance Mishap Query window is displayed revealing each instance (mishap) of the selected description. In addition, the user may page through all mishaps of the selection by selecting the right arrow on the bottom of the window. The data for each mishap is displayed in text boxes, with the selected category denoted with blue background. Additionally, maintenance related contributing factors to the mishap with their HFACS-ME codes are displayed at the bottom of the window. At any time the user may close a window and return to the previous sub-menu by selecting the "Close Form" or "< Back" control. Each of the primary menus has a "Return to Main Menu" control which returns the user to the Main Menu when selected.

The query menu also contains control buttons that allow queries based

on “Multiple Criteria” and multiple “HFACS-ME Elements”. Selecting the “Multiple Criteria” control on the Query Menu allows the user to select any or all of the seven solo categories. A Multiple Criteria sub-menu appears and allows the user to “check” the desired categories and further define them by selecting criteria provided in combo boxes on the sub-menu. A summary of all mishaps sorted by HFACS-ME factors is also available.

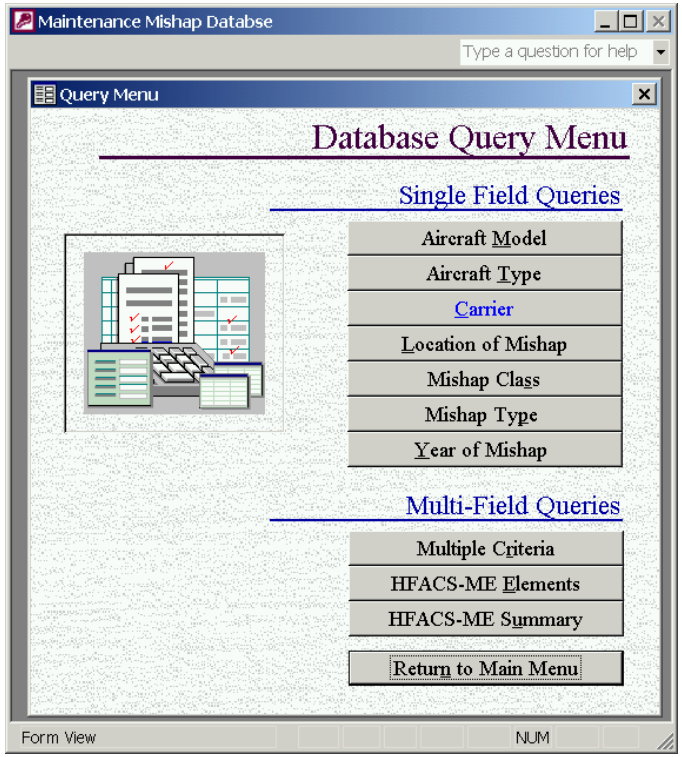

Figure 3: MEIMS Query Menu

The second option from the Main Menu is the Graph Menu. The user can create a two-axis, three-dimensional graph presentation. The x- and y-axes are populated with one of the following categories: aircraft model, aircraft type, mishap location, branch of service, mishap class, mishap type, HFACS-ME level one, HFACS-ME level two, and HFACS-ME level three. The user may then select one or more of the fields from each axis category sub-menu via a combo box. The resultant graph is presented in a three-dimensional, multi-colored view.

The Reports Menu is the third option a user may select from the Main Menu. The Report Menu offers eight reports which provide data listing the total number of mishaps and the number and percentages of mishaps by HFACS-ME levels one, two, and three. The user may select from the following distribution presentations: all mishaps, aircraft model, mishap class, mishap type, and mishap class by mishap type, branch of service, mishap location, and chronological listing by aircraft model. All reports are closed by selecting the “Close” control at the top of the window. Add New Mishap is the fourth option in the main menu. This option allows the user to add data directly to the database tables.

The Mishap Investigation option takes the user to a decision support system that is designed to aide the investigator in determining the mishap cause factors as well as entering the required mishap data.

The Administration option is a password protected entry into the database. Administrators can directly edit, add or delete records. Several tools are available to manipulate viewing and editing records. Administrators may also update the entire database including stored procedures and all tables via an FTP connection or CD-ROM. The ability to switch between the civilian and military version is also supported via drop-down menu.

In summary, this new MEIMS is a true client/server application with the client containing the business rules able to connect to a database server on the desktop, through an intranet connection or even through the Internet. This architecture allows MEIMS to be used on a laptop, within and organization or across geographical boundaries. The intranet paradigm is especially attractive for sensitive data that can be housed in one database location within the organization. The two-tier client/server architecture also sets the stage for development of a web-based three-tier client/server system.

3. HFACS-ME Web

A significant drawback to the original MEIMS tool was that it ran as a stand-alone application with the underlying database stored locally on the host computer. While this model provides for responsive data queries, analysis, report generation and local data entry, it did not support data import/export to a centralized database. Consequently, the data contained in the original MEIMS tool could be out of date before it was received by the end user. Newly entered

or modified data remained unavailable to end users until the MEIMS application was updated. To update data, authorized users had to contact the Naval Safety Center (NSC) to obtain recent mishap information. NSC technicians then manually key in the search criteria. After the query was executed the results were forwarded to the requestor via email or hardcopy. This process could take several hours to weeks, depending on the query backlog and the requestor's location. To update MEIMS, the user would then manually enter the new data. Even with the new client/server version of MEIMS, obtaining updated data would use the same process.

The data currency problem can be avoided by using an enterprise-level database management system (DBMS) that is updated by the Naval Safety Center upon completion of mishap investigations. This DBMS could be tied to a web server. In this three-tier client-server environment, any user with a web browser can query the enterprise-level DBMS by navigating to the Uniform Resource Locator (URL) or "address" of its web server. A Hypertext Markup Language (HTML) page is returned that gives the user the ability to select options from drop-down lists, text input boxes and menu selections. The query criteria are then transmitted to the web server which contains the business logic and database connection information. The web server processes the appropriate HTML page or Active Server Pages (ASP) and forwards the request for data to the database server. Once the database server returns the corresponding raw data, the web server formats the data according to the specification of the called page and returns the results to the browser. The total time to process the custom query and display the results to the user is measured in seconds rather than hours. The end user is able to access the HFACS-ME database in a fraction of the time and obtain up-to-date information that can then be used in training, hazard identification, and trend analysis to prevent possible future incidents. The resulting application takes advantage of a centrally managed and secured database while providing the ability to make the information available to the greatest number of authorized users.

The Human Factors Analysis and Classification System–Maintenance Extension (HFACS-ME) Web application is a prototype based upon web technology [1]. The underlying web pages were designed using FrontPage 2000, and are hosted on an Internet-connected web server running Microsoft Internet Information Server (IIS) 5.0. The GUI for the system is provided by a web browser and allows an end-user with basic computer skills to operate the system. The HFACS-ME data and business logic are contained in a Microsoft SQL Server 2000 in the form of stored procedures and views. The prototype (which contains information on 595 maintenance error related mishaps that occurred between 1989 and 1999) utilizes data derived from the NSC's Safety Information Management System (SIMS) database. The redesigned version of MEIMS and the HFACS-ME Web prototype were developed to use the same database. This provided a cooperative environment to develop the database schema and stored procedures. The end result was a two-tier client server application (MEIMS) and a three-tier client server application (HFACS-ME Web) that used the same database. In one test scenario we simultaneously used the same SQL Server 2000 DBMS to connect to an intranet based MEIMS and HFACS-ME Web. This scenario made it easy to keep the data current on both applications.

The use of stored procedures dramatically improved the performance of HFACS-ME. Without stored procedures, strings of SQL code had to be sent to the database server. For complicated tasks, this code became very large and measurably slowed the request process. The same request could, in some cases, be handled with just a call to execute the stored procedure within the database which would then return a small Recordset.

The HFACS-ME Web prototype was designed to allow the user to access the database via four functional areas: (1) mishap data presentation and filtering, (2) HFACS-ME factor analysis, (3) graphical presentation of multi-dimensional data, and (4) Factor Distribution Report generation. Each function is displayed on separate pages with interactive controls providing the user interface.

The Homepage of the HFACS-ME Web prototype (Figure 4) is the initial page the users see after entering the appropriate URL in the address bar of their browser. Menu selections (hyperlinks) allow the user to select one of four main options: (1) Mishap Data, (2) Factor Analysis, (3) Graph Data, and (4) Reports. Each item can be accessed by clicking on the hyperlink in the menu bar, located below the HFACS-ME logo. Help is provided to the user on this and all pages in the form of "tool tips" (i.e., brief description) when the mouse arrow is placed over a control (e.g., hyperlinks, images, etc.). There are also links to support pages for an Online Tutorial, Help page, Access Policy page, and a Contact Us page.

The Data Selection Page is common to the Mishap Data, Factor Analysis and Graph Data sequences. On this page the user can use nine drop-down boxes to select what data is to be included in the dataset returned from the database. Each drop-down box has a selection of (All). When (All) is the selected item, no criteria are passed for that category to the SQL stored procedure.

The first six drop-down boxes are related to the mishap event itself, and they are: Aircraft Type (e.g., A4, F14, H53, P3, etc.), Mishap Type (i.e., Flight Mishap (FM), Flight-Related Mishap (FRM), and Aircraft-Ground

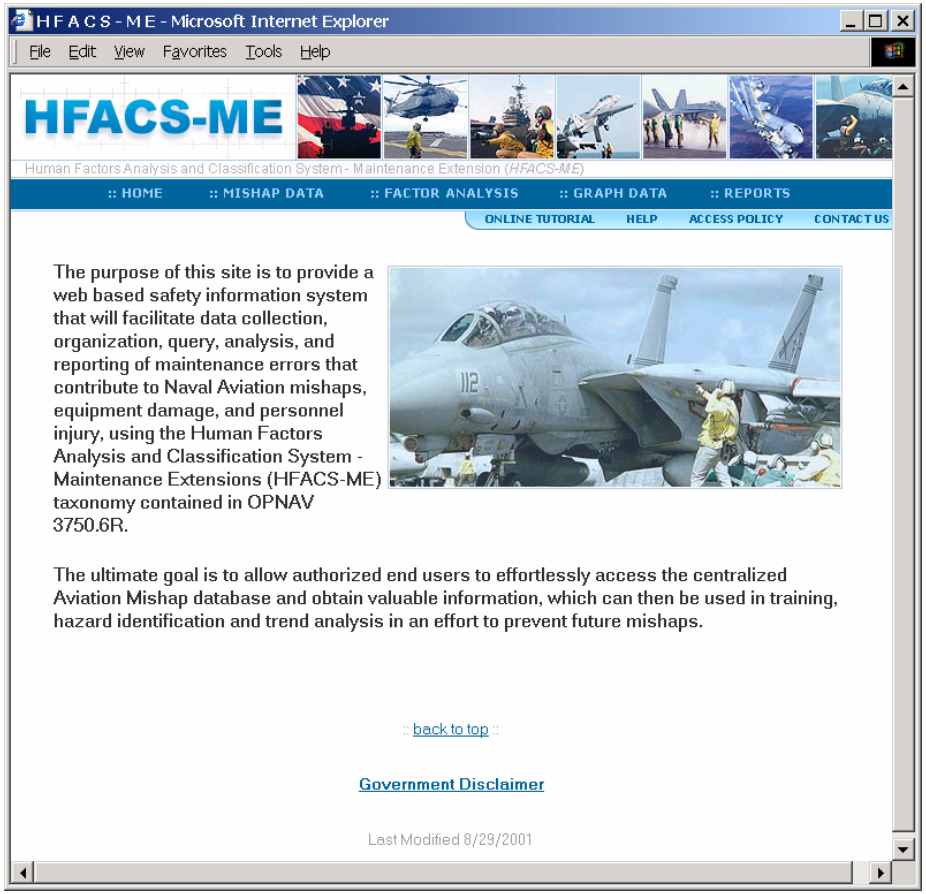

Figure 4: HFACS-ME Web

Mishap (AGM)), Mishap Class (i.e., A, B, and C), Location (e.g., Ashore, Embarked, Detached, etc.), Service (e.g., USN, USMC, etc.), and Fiscal Year (i.e., 1990-1999).

The final three drop-down boxes are related to HFACS-ME taxonomy factors that were present in a given mishap, and they are: 1st Level Factors (e.g., Management Conditions, Maintainer Acts, etc.), 2nd Level Factors (e.g., Crew Coordination, Medical, etc.), and 3rd Level Factors (e.g., Adverse Mental State, Attention/Memory, etc.). Instructions are included to guide the user in selecting multiple items within a particular category, and an explanation of how the criteria are used within the query is given, For example:

> Use the drop down boxes below to select the desired Mishap Criteria and Factors Criteria to include in the dataset. Multiple items <u>within</u> a particular criteria group (i.e. Aircraft Type = F14, F18) may be selected by holding down the Ctrl button and left clicking on the desired items. This will result in a query that will return data matching ANY of the selected items.
>
> If criteria are selected in several criteria boxes (i.e. Aircraft type and Mishap Class), the resulting dataset will be those records that match ALL criteria. For example, if the user selects *Aircraft Type=F18, F14* and *Mishap Class=A*, then the database will return all records involving either a *F14* or *F18* and resulted in a *Class A* mishap.

When the user has completed making all criteria selections, they click on the Submit button at the bottom of the page. This *on_click* event will fire the Submit event in the ASP code and submit all data from the nine drop-down boxes (and any hidden form values) to the next page. The Reset button allows the user to return all values to their original state when the page was first loaded. The Back button is equivalent to the user clicking on the browser Back button, in that it loads the previous page from the browser's History.

The Mishap Details Page displays the list of mishaps matching the user-selected criteria. The total number of mishap matching the user's criteria is displayed at the top of the table. The mishap data is displayed in a table view that contains the following columns: Mishap ID, Aircraft, Type, Class, Location, Service, and Date. Even though the HFACS-ME Factors criteria (i.e., 1st Level, 2nd Level and 3rd Level Factors) are used to filter the dataset, the factors themselves are not included in the table display. The Mishap ID is a hyperlink (blue text and underlined are standard to indicate the presence of a hyperlink), which when clicked on will load the Expanded Mishap View to display greater mishap and factor details.

The Expanded Details Page displays information specific to a single mishap. In addition to the basic mishap data displayed on the Mishap Details page, the Expanded Details page provides a mishap summary and a list of maintenance related factors identified in the mishap. For each factor identified, a factor summary and the 1st, 2nd and 3rd level HFACS-ME taxonomy factors associated with each are displayed. There is also a checkbox that allows the user

to select verbose descriptions vice summaries to be displayed on this page. To change from summaries to long descriptions or back again, the user clicks on the checkbox and then on the Refresh button below it. The checkbox label changes based on the current display setting (i.e., when displaying summaries the label reads “Display Long Descriptions” to indicate the change that would take place if the checkbox is selected. When long descriptions are being displayed the label reads “Display Summaries”).

The Factors Analysis page displays a composite view of the HFACS-ME factors represented in the database. When the page initially loads, the table displays the factor distribution across all mishaps contained in the database. The user can filter the data represented in the Factor Analysis table by using the drop-down boxes below the table (as previously described in the Data Selection page) The table is divided such that the leftmost column displays the 1st Level Factors, the second column displays the 2nd Level Factors, and the remaining columns display the 3rd Level Factors.

1st Level Factor	**2nd Level Factor**	**3rd Level Factor**			
Mgmt Conditions 49 75%	**Organiza-tional** 215 36%	**Inadequate Processes** 94 16%	**Inadequate Doc.** 91 15%	**Inadequate Design** 77 13%	**Inadequate Resource** 4 1%
	Supervisory 335 56%	**Inadequate Supervision** 259 44%	**Inapprop-riate Operation** 78 13%	**Uncorrect-ed Problem** 33 6%	**Super-visory Misconduct** 91 15%

Figure 5: Factor Analysis Layout

Figure 5 illustrates that the 1st Level Factor-Management Conditions has two 2nd Level Factors-Organizational and Supervisory. The 2nd Level Factor Organizational has four 3rd Level Factors: Inadequate Processes, Inadequate Documentation, Inadequate Design and Inadequate Resource. Each cell in the Mishap Analysis table displays the factor name, the number of mishaps containing that factor and the percentage of the total mishaps that it represents. One important fact to note is that the number of mishaps listed for each of 1st Level Factor cannot be determined merely by adding up the number of mishaps listed in the 2nd Level Factors column. This is because multiple factors often are identified for each mishap. When two or more factors fall under a single higher-level factor, the higher-level factor is incremented once for each mishap and not for each factor, to avoid duplicate counts from skewing the higher-level factors.

Graph Page - When a user selects the Graph Data menu hyperlink, they are presented with a Graph Data Selection page, much like the Mishap Data Selection page. Once the user selects the criteria to be included in the graph and clicks on the Submit button, the Grouping Select page loads and allows the user to select two of the three-axes (the third being the Factor Count, which is pre-selected) to be displayed on the subsequent Graph page. The possible selections for each axis are: Aircraft Type, Mishap Type, Mishap Class, Location, Service, Fiscal Year, 1st Level Factor, 2nd Level Factor or 3rd Level Factor. Once the axes criteria have been selected the user has the option to select Display Graph or Display Graph and Data Table. Both options open the resulting Graph page and the latter option opens a second window that displays a color-coded table of the graph data. The number of mishap factors matching each of the criteria selected by the user is presented in a three-dimensional, multi-colored view. To aid in identifying specific data, each vertical column displays the values of each of its three axes when the user moves the cursor over any portion of the column. Due to the display size limitation, all graphs where HFACS-ME Factors (i.e., 1st, 2nd or 3rd Level Factors) are selected to display on an axis, display factor codes vice the long factor name. To assist the user in understanding the codes, a hyperlink, which opens an HFACS-ME taxonomy legend, is displayed next to the graph title.

Reports Page - Clicking on the Reports hyperlink on the menu bar loads the Report Menu page. On this page the user is presented with six report options. Each report details the number and percentages of mishaps by HFACS-ME 1st, 2nd and 3rd Level factors. The user may select from the following distribution presentations: Aircraft Type, Mishap Class, Mishap Type, Location, Organization and Fiscal Year. The resulting Report displays individual reports for each subgroup within the presentation category. Hyperlinks are displayed throughout the report to allow the user to move easily from one report to another.

Support Pages - Additional pages are provided to assist the user during their use of the website or to provide information about the site. Presently the Help and Online Tutorial pages are not operational. The Help and Online Tutorial menu items were added to the website menu bar as "placeholders" in anticipation of future development in the prototype.

Web-based benefits come at a cost. HTML does not produce as rich a user interface as many two-tier client/server programming languages such as VBA. Dynamic HTML and XML may produce improvements to the web

interface and will be studied in future versions of HFACS-ME .

There are numerous challenges that must be overcome to ensure that the data upon which the HFACS-ME Web Site is built remains secure. Mechanisms must be in place to authenticate users before allowing access to the data. Once authenticated, a user should not have to re-authenticate during a single session. This challenge stems from the "stateless" nature of the Hypertext Transport Protocol (HTTP) server, which view all page requests as independent and therefore does not maintain access credentials as the user moves from page to page by default. Data security/integrity as it traverses the Internet is also of concern given the open nature of the Internet. To ensure data is not viewed by unauthorized personnel or tampered with in route, security measures such as secure socket layer (SSL) may be in order. Other issues unique to the web implementation deal with concurrency and scalability. Concurrency deals with the problems resulting from two or more end users modifying the same data at the same time. In the current prototype there is no data input, deletion or modification implemented, so this issue does not pose a problem – yet. Scalability, on the other hand, has to do with the Web server and database server's ability to process hundreds, thousands, possibly millions of requests for data in a given period of time.

In summary, HFACS-ME Web fits in well with the global mission of the Navy. And the future growth of web-based capabilities provided by the Navy Marine Corps Intranet (NMCI) and IT-21 will provide a logical path for the Naval Safety Center to provide access to the Aviation Safety database. This vision is echoed in the Web Enabled Navy Architecture, dated 31 January 2000, where Admiral W. J. Fallon sees "A Navy in which operational and business processes are conducted worldwide via interconnected and interoperable Web-based IT systems."

4. Future Enhancements

The real test for the system depends on whether the system adds value to the underlying mishap data, and ultimately, whether the end user gains knowledge that leads to effective intervention and mishap prevention. This hinges on the ability of the user to conduct iterative searches on the mishap data, analyze past maintenance mishap factors, extract trends and construct predictors of future problems. A pilot study found that many potential users were unfamiliar with query techniques or trend analysis and could not use the system to its full potential. Therefore, training in these areas is considered critical.

Since many of the potential users of organizational knowledge bases will be organizationally and geographically distributed, conventional classroom methods for instruction may not be feasible. Tacit knowledge development by organizational socialization will also be difficult. An on-line tutorial is being developed to help users become familiar with MEIMS and HFACS-ME Web as well as some basic database query techniques. A web-based Asynchronous Distributed Learning (ADL) module is also being considered for development because it can give users the ability to learn at their own pace in an interactive environment [3].

MEIMS and HFACS-ME were designed to help users analyze past mishaps. It is anticipated that individual users will investigate aircraft specific mishaps and perhaps specific mishap factors but it is not anticipated that any individuals will take a holistic approach to analyzing the data. A detailed analytical approach could provide the biggest potential for mishap prevention. Data-mining could be useful for this process. Data-mining is the non-trivial extraction of implicit, previously unknown, and potentially useful knowledge from data. It uses a variety of techniques to identify nuggets of information or decision-making knowledge in bodies of data, and extracting these in such a way that they can be put to use in areas such as decision support, prediction, forecasting, and estimation. In recent years, specific tools have been designed to support data-mining. In fact, the overriding decision for choosing the Microsoft SQL Server 2000 DBMS was to take advantage of these tools. Therefore, because of its great potential, data-mining aviation mishap data for clustering patterns, statistical pairings and cause and affect relationships will be the focus of our future research efforts.

5. Conclusions

Mathematical analysis of the HFACS-ME taxonomy has shown it to be viable model for discovering trends in aviation mishaps. Initial pilot testing of earlier versions of MEIMS has shown it to have great promise. The challenge was to reengineer an earlier version of MEIMS, redesign its database model and create a web-based version to enable distributed database-mining of current aircraft mishap data. The enhanced desktop version and the new web-based version meet these requirements. When they are used with the upcoming ADL module, and data-mining tools, MEIMS and HFACS-ME Web will comprise a complete, robust system for analyzing aircraft maintenance mishap related factors anywhere at anytime.

6. References

[1] Boex, A. R., *Web-Based Information Management System for the Investigation, Reporting, and Analysis of Human Error in Naval Aviation Maintenance*, Master's Thesis, Computer and Information Systems Department, Naval Postgraduate School, Monterey, CA (2001).

[2] Department of the Navy (DON). *Naval Aviation Safety Program, OPNAVINST 3750.6R, Appendix O*. Washington, DC: Office of the Chief of Naval Operations (2001).

[3] Dijkstra, S., N.M. Seel, F. Schott, and R.D. Tennyson. *Instructional Design*, Lawrence Erlbaum Associates: Mahwah, N.J. (1997).

[4] Flanders, T. P. and S. K. Tufts, *Software Re-engineering of the Human Factors Analysis and Classification System – (Maintenance Extension) Using Object Oriented Methods in a Microsoft Environment*, Master's Thesis, Computer Science Department, Naval Postgraduate School, Monterey, CA (2001).

[5] Fry, A.D., *Modeling and Analysis of Human Error in Naval Aviation Maintenance Mishap's,* Master's Thesis, Operations Research Department, Naval Postgraduate School, Monterey, CA (2000).

[6] HFQMB, Human Factors Quality Management Board Charter. http://www.safetycenter.navy.mil , (1997).

[7] Naval Safety Center (NCS), Aviation Safety Data, http://www.safetycenter.navy.mil/Statistics/aviationst1.htm , (2001).

[8] Nutwell, R., and K. Sherman, *Changing the Way We Operate,* Naval Aviation News, March-April, 79(3), Washington, DC (1997).

[9] Schmidt, J., *Human Error in Naval Aviation Maintenance,* Presentation at the Joint Service Safety Chiefs Aviation Conference, Ft Rucker, AL (1996).

[10] Schmidt, J., Schmorrow, D., and M. Hardee, *A Preliminary Human Factors Analysis of Naval Aviation Maintenance Related Mishap,* (983111), Society of Automotive Engineers, Inc (1997).

[11] Schmorrow, D., *A Human Error Analysis and Model of Naval Aviation Maintenance Related Mishaps,* Master's Thesis, Operations Research Department, Naval Postgraduate School, Monterey, CA 1998).

[12] Wood, B.P., *Information Management System development for the Characterization and Analysis of Human Error in Naval Aviation Maintenance Related Mishaps,* Master's Thesis, Computer and Information Systems Department, Naval Postgraduate School, Monterey, CA (2000).

Biographical Sketch

George Zolla is a lecturer and IT PhD student at the Naval Postgraduate School teaching Client/Server architecture, programming and computer operating systems. He is a retired U.S. Navy Captain with extensive experience in tactical aircraft and shipboard operations. He has had numerous assignments in aviation maintenance and has served as the senior member of an aircraft mishap board. His education achievements include a Bachelor of Science in Aerospace Engineering from the Pennsylvania State University and a Master of Science Degree in Information Systems from the Naval Postgraduate School. Recent publication topics include Expert Systems, intranet development, information technology diffusion, and distributed maintenance error information.