

Final Report
On the project of
Digital Form Processing with XML

FAA Grant Number 00-G-031

(October 1, 2000 to October 30, 2003)

Submitted to
Office of Research and Technology Applications
FAA Tech Center

On

January 29, 2004

By

Dr. Mou-Liang Kung
Department of Computer Science
Norfolk State University
Norfolk, Virginia 23504

(757) 823-8650 (voice)

(757) 823-9229 (fax)

mlkung@nsu.edu

TABLE OF CONTENTS

Topic	Page Number
I. PROJECT SUMMARY	3
II. PROJECT DESCRIPTION	4
III. DESCRIPTION OF WORK ACCOMPLISHED	9
1. Grant Administration	9
2. Information Gathering and Dissemination	10
3. Research Activities	12
4. Future Plan of Work	12
IV. DELIVERABLES	12
V. APPENDICES	13
A. User Guide For the FAA MIS/SDR Form Reporting System	13
B. Installation Guide For the FAA MIS/SDR Form Reporting System	34
C. Developing a Standalone Web Service Client Accessing HAI Web Service	42
D. Two winning Student Presentations: Virginia Academy of Science Annual Meeting, and National Technical Conference	43
E. “Dynamic Multi-modal Multi-Platform XML Publishing - A student paper	44
F. Java-based client/server system with a form to generate XML documents - A student paper	83
G. Web Form Page to Generate XML Document from User Input - A student paper	88

I. PROJECT SUMMARY

The primary goal is to develop and demonstrate an extensible Web-based information system for aircraft maintenance to improve commercial aircraft safety. Primary objectives are to allow 1) the FAA to automate the submission, storage and retrieval of the aircraft maintenance data from the Mechanical Interruption Summary (MIS), and the Service Difficulty Report (SDR) that FAA requires, and 2) fast data sharing with other FAA information systems. The benefits of such a system to FAA and the aviation industry include the reduction in labor required for aircraft operators to submit electronic data, reduction in labor required for the FAA to collect, maintain, and disseminate comprehensive data, for safety analysis and trend detection via the Web, and fast data sharing between the FAA and industry using the platform-neutral XML data format and web services

We also built a proof of concept XML-based, Multimodal Publishing System (MUMPS) based on software such as Tomcat/Cocoon, IBM Voice Toolkit, MySQL and Xindice. The multiple modality publishing capability permits universal access (any time, any place, any communication means, and for the physically challenged) from multiple wired or wireless devices. This proof of concept demonstrated how an information system can be made extensible for future technological advancement.

II. PROJECT DESCRIPTION

The Problem

In the Safer Skies initiative, the FAA administrator promotes the partnership between the FAA and the aviation industry in improving aviation safety. Partnership programs include ongoing analytical program with industry to determine the root causes of accidents. Once root causes are identified, intervention strategies will be developed to address the problem. The FAA also embarks on aviation data improvements including quality, collection, and analysis. Rapid development in software tools and data processing, plus the effort to build partnerships with the aviation industry, give the FAA analytical tools and institutional relationships that did not exist. The FAA and industry now have a new way to prevent accidents, including accidents with new scenarios that cannot be reasonably anticipated with previous analytical tools.

The Federal Aviation Regulations (FAR) requires aircraft operators to submit the Mechanical Interruption Summary (MIS), and the Service Difficulty Report (SDR) to the FAA as follow:

Service Difficulty Reports – Each scheduled operator is required under FAR Part 121.703 to report the occurrence or detection of each failure, malfunction or defect concerning (a) fires during flight major, (b) false fire warning during flight, (c) engine exhaust system that causes damage during flight, (e) an aircraft component that causes accumulation or circulation of smoke, vapor, or toxic or noxious fumes during flight, (f) engine shutdown during flight, (g) a propeller feathering, (h) aircraft structure requiring major repairs, (i) cracks, corrosion, (j) other safety critical issues as stated in the FAR part.. These occurrences must be reported within 72 hours of the event.

Mechanical Interruption Summary – Each scheduled operator is required under FAR Part 121.705 to submit a summary of any (a) interruption to flight, (b) unscheduled change of aircraft en route, or unscheduled stop or diversion from a route caused by known or suspected mechanical difficulties or malfunctions that are not reported as service difficulty reports.

However, there is no system in place to automate the collection, storage, and retrieval of the submitted data. Therefore, this safety critical data is not readily available for the FAA and aviation industry to perform safety analysis and trend detection. Also, as many of the required data elements in the MIS and SDR forms are duplicated, it is an unnecessary burden for aircraft operators to comply with the FAR.

Technical Approach to Our Solutions

With the proper design of Web-based information system and proper implementation of digital form processing, many benefits can be reaped: cost saving, accurate record keeping (no error-prone human intervention), process automation, etc. However, there are many incompatible formats ranging from proprietary word processing formats (e.g. Microsoft Word) to other form formats such as PDF, PDF-F, JetForm, InForm, etc. The popular HTML (using FORM tag) format is completely inadequate for complex forms. It is also recognized that “the next generation forms are incompatible with preserving full backwards compatibility with browsers designed for earlier versions of HTML” (<http://www.w3.org/TR/xhtml-forms-req>). Therefore, the development of Web-based information system and implementation of digital form processing need to be carefully designed to avoid data obsolescence due to proprietary formats.

Unfortunately, the current model of using Hypertext Markup Language (HTML) web pages to transport data between a web browser and a web server to a database front-end failed to meet any of the requirements above due to:

- No data validation – Business data or documents typically contain data component of different types. For HTML documents, there is no **native** way (e.g. via JavaScript programs) to check the validity of data or the correctness (syntax) of data before they are entered into a database. This makes stored data less useful for retrieval or data analysis. HTML hyperlinks are also primitive and hard to manage. It becomes a daunting task to validate data containing references to other data across the Internet.
- No data reusability – Since HTML tags are non-extensible, unstructured non-hierarchical data entered into an HTML web page is completely unusable for any other purpose at the client and making any effort of automated processing impossible.
- No open data interchangeability – When the data enter a database, the data become records of specific record format (e.g. Microsoft Access). This makes data exchange with heterogeneous systems impossible without data conversion programming effort.
- No data longevity – Data is always captured in a platform or application dependent format, which makes it harder to foster business process automation. The application now owns the data rather than the creator of the data. As the application goes, so goes the data.

However, there is an open document exchange standard today that can replace the HTML without its data processing deficiencies: the XML (eXtensible Markup Language). XML is a powerful subset of SGML (Standard Generalized Markup Language, ISO 8879) tailored for Web deliveries. SGML and XML are being adopted as the open document exchange standard in various federal agencies as well as industrial and commercial enterprises. XML meets the business requirements that HTML failed on the following functionality:

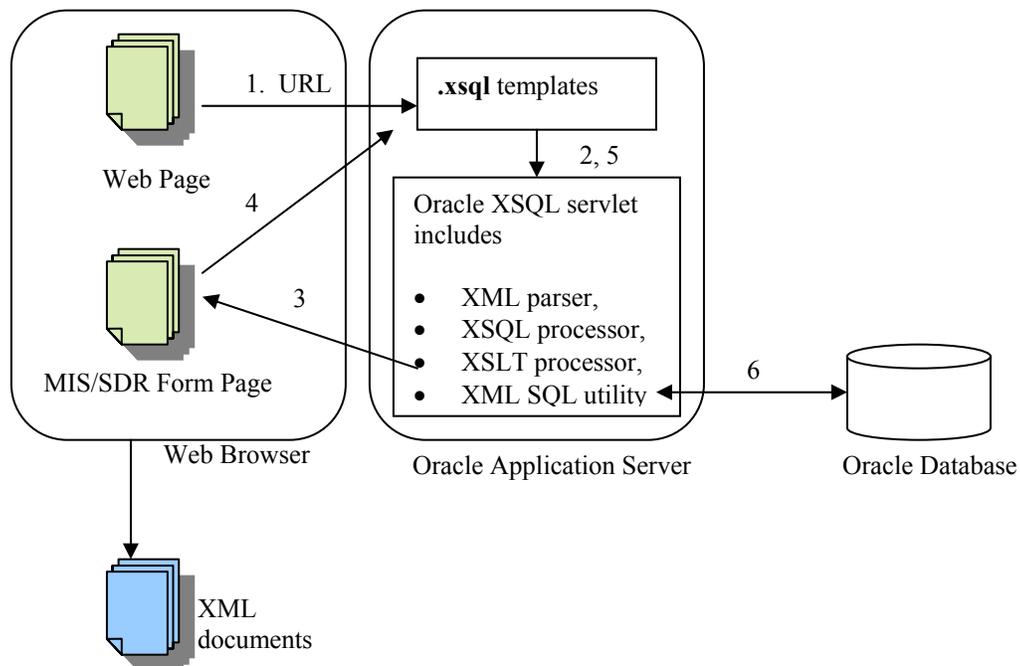
- Data Validation – Data structural syntax can be enforced with XML parser, and data type validity can be checked.
- Data Reusability – Data hierarchical and structure are native to XML and they can be preserved without any loss of data intelligence.

- Data Interchangeability – XML standard is platform and application independent
- Data Longevity – XML, a meta-language, is open and extensible, not restricted to a single tag set like the HTML
- Data Security – The hierarchical structure in XML documents also intrinsically permits easy security enforcement with the digital signature specification for XML and component-wise data encryption. XML also permits finer grain of access control to render a document in different appearances for different users in full text or abridged form (with sensitive text hidden). This provides data confidentiality at no additional cost, since it would allow a document created once but published many times.

The NSU researchers recognized that even the most technological advanced information system would not benefit its users – both the FAA and aviation industry – unless the system was developed with the users’ needs and requirements as priority. To this end, NSU has enlisted the assistance of a subject matter expert (SME) in aircraft maintenance. This SME complemented the NSU researchers’ skills in information technology by consulting the research team on aircraft maintenance issues. The SME helped the NSU researchers to understand and identify the users’ requirements as well as the process of collecting and disseminating aircraft maintenance data. The SME also assisted the NSU researchers to understand the pertaining FAR and interact with FAA organizations and aircraft operators for data and inputs during course of system development and demonstration.

Our MIS/SDR SYSTEM Design

The following diagram illustrates our design of the MIS/SDR digital form processing system built in which, Oracle 8i/9i provided many tools that can be readily used without doing much programming. In particular, Oracle uses XSQL Pages (http://otn.oracle.com/tech/xml/info/htdocs/otnwp/about_oracle_xml_products.htm), which are server-side XML templates, to exploit the powerful combination of SQL, XML and XSLT to prototype and deploy dynamic, data-powered sites and web services.



Personnel

Personnel conducting the work consists of a Principal Investigator (PI): Dr. Mou-Liang Kung), a Faculty-Associate (FA): Ms. Deborah Chen, an Instructor for Database courses and two student Research Assistants (RA) over a period of 3 years from October 1, 2000 through September 30, 2003.

The PI managed the project and followed the description of this plan of work. The PI also served as the point of contact for the project to interact with the FAA Technology Center personnel. The PI and the FA were assigned with primary responsibilities for each area of the investigations. Both are responsible to produce reports and system demonstration.

Two student RAs were selected by the PI each academic year and summer. They were mostly promising students who are interested in pursuing research work after they finish college. The RAs were trained by the PI and the FA to perform the document search and compilation, applied critical analysis skills to judge the scientific values, and extract essentials from the documents. The RAs also gave presentations on their findings in departmental seminar and conferences.

Since the subject pertaining to aircraft system maintenance is very specialized, we solicited Mr. Frederick J. Leonelli of the FJLG Group, Inc. as our Subject Matter Expert (SME) to provide consulting support on the operations of Mechanical Interruption Summaries (MIS), and Service Difficulty Reports (SDR). The consulting effort is estimated at 400 man-hours during the first year, 200 man-hours the second year and 70 man-hours the third year.

Milestones

- Year 1** October 1, 2000 - September 30, 2001
Demonstration of a properly working system implementing MIS digital form processing system
- Year 2** October 1, 2001 – December 31, 2002
Modified MIS system according to input from FAA inspectors and airline maintenance personnel; Demonstrated a properly working combined MIS and SDR form processing system
- Year 3** January 1, 2002 - October 30, 2003
Modified the combined MIS and SDR form processing system. Added Web Services capability to the System; Demonstrated an interoperability with HAI or other Web Services; Demonstrated multimodal publishing to multiple platforms including desktop, PDAs, mobile phones, and traditional telephone handsets.

III. DESCRIPTION OF WORK PERFORMED

The activities conducted during this project include the followings:

1. GRANT ADMINISTRATION

- Faculty-Associate (FA): Ms. Deborah Chen was retained throughout the grant period. Mrs. Chen, an Instructor for Database courses, brought her expertise in database to this project. Tasks assigned to her were all accomplished on time and compliant to the specification details.
- 2 student Research Assistants (RA) were hired each semester. They received stipends at \$8.50/hr for 10-hour work weeks during academic years and 40-hour work week in summers. Not only they contributed to the project, they were also trained to pursue a research career in the future.
- Several notable student Research Assistants were hired:
 - Ms. Cyntrica Eaton, a top graduating senior, has graduated and went to University of Maryland to pursue her graduate study in computer science
 - Mr. Ian Bradley, graduated and employed by NSU Office of Information Technology. He is currently enrolled in NSU computer science graduate program as well.
 - Ms. Jacquelynn Dolly (a senior), and Mr. Moses C. Alltey-Pappoe were hired in the Fall of 2002. Within a short 3 months period, they have added new capability to the Tomcat/Cocoon server by publishing HTML pages dynamically generated from XML documents. They won the First place in student presentation in two conferences: 2003 Annual Virginia Academy of Science Meeting, and 2003 Annual National Technical Association meeting. They both have aspiration to pursue graduate study when they graduate.
- We solicited Mr. Frederick J. Leonelli of the FJLG Group, Inc. as Subject Matter Expert (SME) to provide consulting support on the operations of Mechanical Interruption Summaries (MIS), and Service Difficulty Reports (SDR), the Safety Performance and Analysis Subsystem
- Weekly progress meetings were held among NSU team members, typically every Friday at 11 A.M.
- Throughout the project, the System we built went through several revisions and upgrades from Oracle 8i database server to 9i.
- Laptops, digital electronic and peripherals (e.g. PDAs, DVD drive, scanner, USB hub, networking hubs, wireless networking devices, audio and video recording devices, LCD projector, ...), software and supplies were purchased to support various project related research activities and demonstrations.

2. INFORMATION GATHERING AND DISSEMINATION

Meetings and Briefings

- Dr. Kung met with Subject Matter Expert (SME) Mr. Frederick J. Leonelli of the FJLG Group, Inc. on Oct 31, 2000 at Holiday Inn, RIC Airport, Richmond, and again on December 7 and 8, 2000. The meetings clarified
 - a. Details and flow of SDR and MIS reports. It is quite important for us to learn the similarity among them for the design of the digital form processing system, even though SDR and 337 will come later in years 2 and 3
 - b. Details of Form 337. It is still unclear as to how to deal with large volumes of attachments. A trip to FAA data processing center at Oklahoma City will be planned.
- On Dec 7, 2000. we visited HAI and made a progress presentation to Mr. Michael Vu, our FAA Point of Contact, and demonstrated two graphic user interfaces built to create XML documents from user input (screen capture and source code included in Appendices I and II)
- On Dec. 8, 2000, the PI met with representatives from Airline Transportation Association (ATA), and National Air Carrier Association (NACA) to get their feedback on the SDR and MIS reports. Later in the afternoon, we visited Atlantic Coast Airlines maintenance depot field operators and observed how MIS reports were created and filed.
- On November 14, 2001, the PI visited FAA Tech Center to demonstrate the MIS prototype. Feedbacks from Mr. John LaPoint and Mr. Michael Vu were later incorporated into the MIS system within a month. During the meeting, we are told that Form 337, originally planned should be put on hold until further instructions and that MIS and SDR forms need to be merged.
- On January 15, 2002, the PI visited US Airways and FAA office in Pittsburgh and demonstrated the revised MIS system. People present included: Mr. Raymond F. Dougherty, FAA Airworthiness Inspector, Mr. Harry W. Hoffman, FAA Aviation Safety Inspector, Mr. Robert Barron, US Airways Chief Inspector, Mr. Basil J. Barimo, US Airways Director of Maintenance Quality Assurance, and Mr. Thomas H. Carroll, III, US Airways Manager of Reliability Engineering. The meeting produced further recommendations for modification of the main interface to MIS system. The recommendations were forwarded to Mr. Michael Vu, FAA Project Manager. A report was submitted in the progress report..
- On August 23, 2002, Mr. Fred Lionelli visited NSU to finalize user interface.
- On August 29, 2002, the PI visit FAA data processing center at Oklahoma City, OK. Issues on how to converge MIS and SDR are discussed. If MIS remains as a separate system, the question then is “what kind of the logistics and protocol for collaboration should be followed in order that the MIS system can interface with SDR at Oklahoma City?” The purpose of the meeting is to discuss the following topics:
 1. SDR system design and MIS compatibility.
 2. Incorporation of information on one form.
 3. SDR/MIS Interface with SPAS.

4. MIS incorporation under SASO.
 - Other issues include implementation Planning and testing
 - On May 12, 2003, the PI briefed on the status of MIS/SDR system to Mr. Michael Vu, our FAA Technical Monitor at HAI in Washington, DC. During the meeting, it is agreed that the task to be completed by the end of the grant period should consist of:
 - a. Incorporate HAI web services into the MIS/SDR system to update its table on aircraft registration
 - b. Publish MIS/SDR system database as web services
 - On June 10, 2003, the PI visit HAI in Washington, DC. Mr. Ed DeCampli of HAI demonstrated its web services and reached an agreement how NSU will access HAI's MMIR web service. Subsequently, Mr. DeCampli provided NSU an account number and the URI for the algo.wsdl for the PI to proceed with a test on MMIR web service.
 - On October 20, Mr. Michael Vu visited NSU to be briefed on the completion of the MIS/SDR system with added capability of using web services to access to HAI MMIR database. This important demonstration illustrated how it may be made useful to other FAA maintenance database operators.
 - On December 4, 2003, the last NSU briefing for information dissemination was held at FAA Headquarters in Washington, D.C. with presence of Ms. Jean Watson, Mr. David Soucie, and Mr. Michael Vu to demonstrate the MIS/SDR system and the possibility of extending the access to mobile devices.

PRESENTATIONS

- Dr. Kung presented 5 seminar talks within the computer science department.
- the Dr. Kung present a paper titled "Optimal Storage for XML Documents," at the 2002 Virginia Academy of Science Annual Meeting in Hampton Virginia, on May 24, 2002
- The paper presented by our Research Assistants: Mr. Moses C. Alltey-Pappoe and Ms. Jacquelynn Dolly won the Best Paper Contest for all undergraduates and graduate students in the Computer Science Session of the Virginia Academy of Science Annual Meeting on May 29, 2003.
- Mr. Moses C. Alltey-Pappoe and Ms. Jacquelynn Dolly also presented a paper at the National Technical Association Annual Conference to be held in Orlando, Florida, September 17-20, 2003
- Dr. Kung presented the research problem of the MUMPS and Web security at the Commonwealth of Virginia Governor's Research Panel meeting in Herndon, Virginia on October 6, 2003

3. **STUDENT RESEARCH ACTIVITIES: WEB-BASED DIGITAL FORM PROCESSING SYSTEM**

- Tutorial sessions on XML were given to new NSU RAs
- Various task assignments were formulated and distributed to RAs
- The experimental system students are building is separated from the production system Ms. Chen, FA is leading. Tomcat/Cocoon was chosen as the publishing platform and MYSQL and Xindice were selected as database backend.
- Study was made and results presented at conferences on enabling Tomcat/Cocoon server as a multimodal multiplatform XML-based publishing system
- Study on the best database management system (XML-enabled DBMS versus XML native) to support XML document storage was conducted. Our immediate conclusion was that XML-enabled relational database management system is very suitable for form data-centric documents while XML native databases are suitable for document-centric documents.
- Once the XML-enabled relational database was chosen, studies were conducted on how XML documents are best stored in tables. There are three ways to do that: 1) store the entire XML document in a field as a BLOB. This extreme is simple but it makes the XML document structure invisible to the database system. 2) Create a table for each XML element. This is the other extreme in which XML document structure is preserved but access may be slow since too many table-join operations might be required. 3) A compromise to store selected elements in tables and not to nest tables too deep. Certainly this is the better way to go.
- Students further investigated the issue of enabling web service on Tomcat/Cocoon to exchange XML documents in databases with other web services.

4. **FUTURE PLAN OF WORK**

Funding sources are pursued to continue the work to

- Publish digital form processing system such as our MIS/SDR system to multimodal, mobile devices such as PDAs and WAP phones.
- Start web service security investigation to enhance web publishing system.

IV. DELIVERABLES

A hard copy and electronic copy on a CD are delivered to include: MIS/SDR system Installation Guide, MIS/SDR User's Guide, Student Final paper describing MUMPS configuration, and this Final Report.

Appendix A

A User Guide

For the

FAA MIS/SDR Form Reporting System

(Version 3.0)

Created by
Ms. Deborah Chen
dkchen@nsu.edu
Dr. Mou-Liang Kung
mlkung@nsu.edu

Department of Computer Science
Norfolk State University
Norfolk, VA 23504

(757) 823-9454 (Voice)
(757) 823-9229 (fax)

January 29, 2004

Table of Content

I. Introduction	15
II. Before You Begin	17
III. Overview of MIS/SDR Form Interface	18
1. Query for Detailed Records	19
2. Create a New MIS Report	20
3. Create a New SDR Report	21
4. Generate Report on a Particular Aircraft	22
IV. User Guide for Each Type of User	25
V. XML Documents	33

I. Introduction

This document is designed to enhance understanding of the FAA MIS Form Reporting System developed by NSU under the FAA Grant. The system's functionalities, capabilities, its different users and the users' privileges when accessing data in the database are briefly discussed. Please provide the user with ONLY the documentations for his/hers respective forms. Security of data in the database may be compromised should a user with lower privileges gain access to the forms and the documents prepared for users with higher privileges.

Users

The following table highlights the forms available and their user privileges:

Form	User	Privileges
SuperMISSDR3.fmx	Super user or DBA of the system	Query, Insert, Update, Delete all table data including the user accounts tables
FAAInspector.fmx	Inspectors of FAA	Query, Insert, Update of all tables except the user account tables
FAAAnalyst.fmx	Analysts of FAA	Query all tables except the user accounts tables
MIS4MDepot.fmx	Person(s) who file(s) the MIS Report	Query, Update existing MIS Reports filed by his/her own company. Insert new MIS Reports. Update own company information.

Database Tables

A relational database will be created using Oracle 9i DBMS and table instances will be created in the database to store vital data for the FAA MIS Form Reporting System. There are total of nine tables and three views used in the system. Their functionalities are described briefly in the table on the next page:

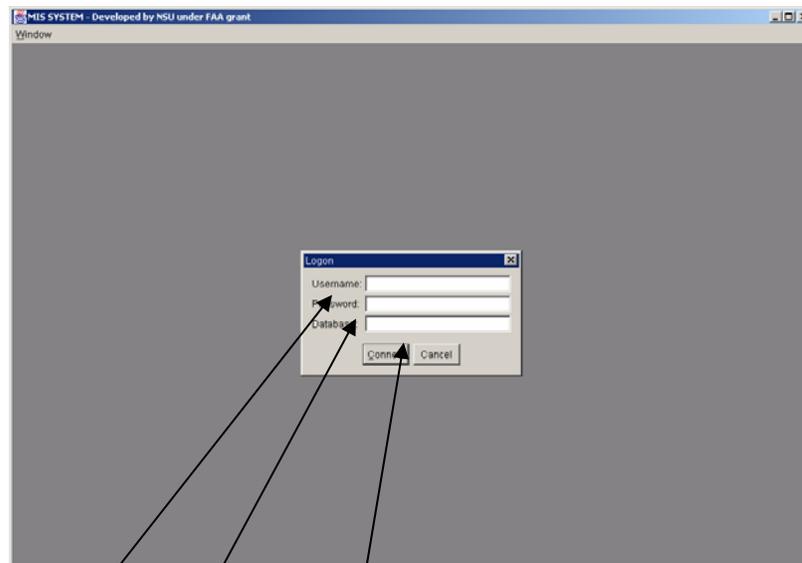
Table / View Name	Description	Accessible to
MIS	A table to store MIS report data filed by maintenance providers	Users at maintenance depots, FAA analysts, inspectors and system administrators.
SDR	A table to store SDR data filed by carriers/airline companies	Users at airline/carrier company, FAA analysts, inspectors and system administrators.
MAINTENANCECO	A table storing basic information about maintenance providers	Users at maintenance depots, FAA analysts, inspectors and system administrators
PART	A table storing vital data about a part	Users at maintenance depots, Users at airline/carrier company, FAA analysts, inspectors and system administrators
AIRCRAFT	A table to store data about an aircraft	Users at maintenance depots, Users at airline/carrier company, FAA analysts, inspectors and system administrator
ENGINE	A table to store basic information about an engine	Users at maintenance depots, Users at airline/carrier company, FAA analysts, inspectors and system administrator
PROPELLER	A table storing information about a propeller	Users at maintenance depots, Users at airline/carrier company, FAA analysts, inspectors and system administrator
MAINTENANCEUSER	A table to store username and name of the user's company	FAA system administrator
AIRLINEUSER	A table storing username and name of the user's airline company	FAA system administrator
INFOONACREG_VU	A view to store selected information from MIS table	Users at maintenance depots, Users at airline/carrier company, FAA analysts, inspectors and system administrator
SUMMARYBYCOMPANY_VU	A view used to generate XML files for selected records	For system internal used only
SUMMARYBYAIRLINE_VU	A view used to generate XML files for selected records	For system internal used only

II. Before You Begin

1. Make sure you obtained a **user-type**, **user login**, **password** and **database name** from MIS database administrator, where user-type can be
 1. Supermissdr3 - Database administrator
 2. FAAInspector - FAA Inspectors
 3. FAAAnalyst – FAA Analyst
 4. MIS4MDepot – End user who submits MIS report
2. You need a web browser (e.g. Microsoft Internet Explorer v.5.0 + or Netscape v 5.0+)
3. There are 4 different URLs for each of the 4 different types of users. In general, the URL to the MIS Form server looks like:
<http://199.111.112.120:7778/servlet/f60servlet?form=user-type.fmx>)
, where “**user-type**” must be substituted by one of the 4 user-type listed (case sensitive) and you need to replace the IP-address by your web server address.
4. All entered data can be saved locally as XML documents (please see part V.).
5. Please report all problems to MIS database administrator

First Time User

1. Start your web browser. Enter the URL to the MIS Form Server in the address field. If the browser prompts you to install **Jinitiator**, please answer **yes** to install and restart your browser later. When you access the same URL, you will see



Now fill in user **login**, **password**, and **database name** (default: **faapart**).

III. Overview of the MIS/SDR Form Interface

The screenshot shows the MIS SYSTEM interface with the following callouts:

- Select MIS or SDR forms:** Points to the left-hand menu where 'MIS' and 'SDR' are selected.
- Use this "Query" to query records:** Points to the 'Query' button in the top menu bar.
- Four record navigation buttons:** Points to the navigation icons (back, forward, first, last) in the top toolbar.
- Click this button to view the entire database records as an XML file after query (See Part 4):** Points to the 'Click to view table contents in XML format' button.
- Click this button to select a record to view as an XML file after query (See Part 4):** Points to the 'Click to select and view a record in XML format' button.
- Click this button to generate a report on a particular aircraft using the Registration No.:** Points to the 'Incident summary by aircraft Reg. No.' button.

Please note: Not all functions are available for all four types of users. Please see **Part IV** for user-specific functions on the interface.

III.1 Query for Detailed Records (for MIS and SDR forms)

Please note that the same form serves as blank form to fill (default) as well as form to query existing records

1. To fetch a known record by the record number: Query -> Enter -> (enter a known number in the Record No. field, e.g. 61) then Query ->Execute to see:

MIS SYSTEM - Developed by NSU under FAA grant

Action Edit Query Block Record Field Help Window

FAA

MIS Record No. 61 Report Time 12/17/02 19:18:51 SDR No. 66

SDR Maintenance Provider RELIABLE ONE Carrier Name WTAA Event Time 12/12/2002

MAINTENANCECO Aircraft Reg No. ACREG1 Model/Make PITTS PITTS S1 Aircraft Serial No. ACSER1

PART

AIRCRAFT

ENGINE

PROPELLER

MAINTENANCE CO. USERS

AIRLINE USERS

Flight Related Data

Flight Num FLT666 Flight Hour 666

Depart Location PROVIDENCE, THEODORE FRANCIS GREY Flight Cycle 6

Depart Time, Schd 12/12/02 12:12:12 Depart Time, Actual 12/12/02 12:12:12

Destination, Schd WHITE PLAINS, WESTCHESTER COUNTY - HPN Destination, Actual WHITE PLAINS, WESTCHESTER COUNTY

Arrival time, Schd 12/12/02 22:22:22 Arrival time, Actual 12/12/02 22:22:22

ATA Code 2866

Engine Related Data

Click in the box(s) next to the problematic engine(s) Engine #1 #2 #3 #4

Engine Serial No. Engine TSI Engine TSO

Engine Total Time

Propeller Related Data

Click in the box(s) next to the problematic propeller(s) Propeller #1 #2 #3 #4

Propeller Serial No. PPSER2 Propeller TSI 66 Propeller TSO 66

Propeller Total Time 132

Event description SLOW TURNING PROPELLER Corrective action PROPELLER #3 REPLACED

Click to view table contents in XML format Click to select and view a record in XML format Incident summary by aircraft Reg. No.

Record: 1/1

To clear all records and return to input mode, click Record -> Clear

2. To fetch all records from the database: Query -> Execute (without doing Query -> Enter first) then use solid right or left arrow key to browse all records.

III.2 Create a New MIS Report

Note:

1. **Current data fields are mostly string types without formats except some date and time fields. Therefore, it would be good to require user to use the following format:**
01-Nov-2000 15:03:00
2. **Please use tab keys to skip over shaded blanks, since these values will be automatically generated.**
3. **Always use tab key to go to the next field; or use Field->Previous/Next to navigate on fields in the same form.**

1. To create a new report: Press “Tab” key over the first shaded blank to generate record number and current time.

The screenshot displays the 'MIS SYSTEM - Developed by NSU under FAA grant' interface. The window title bar includes 'Action Edit Query Block Record Field Help Window'. The main menu bar contains 'FAA'. A left-hand navigation pane lists categories: MIS, SDR, MAINTENANCE CO, PART, AIRCRAFT, ENGINE, PROPELLER, MAINTENANCE CO. USERS, and AIRLINE USERS. The main content area is divided into several sections: 'Record No' and 'Report Time' (with a shaded blank); 'Maintenance Provider', 'Carrier Name', 'Model/Maker', 'SDR No.', 'Event Time', and 'Aircraft Serial No'; 'Flight Related Data' including 'Flight Num', 'Flight Hour', 'Depart Location', 'Flight Cycle', 'Depart Time, Schd', 'Depart Time, Actual', 'Destination, Schd', 'Destination, Actual', and 'Arrival time, Schd', 'Arrival time, Actual'; 'ATA Code'; 'Part Related Data' with fields for 'Part Number', 'Manufacturer', 'Serial Number', 'Nomenclature', 'Part Vendor', and 'Date Installed'; 'Engine Related Data' with checkboxes for 'Engine #1' through '#4', and fields for 'Engine Serial No', 'Engine TSI', 'Engine TSO', and 'Engine Total Time'; 'Propeller Related Data' with checkboxes for 'Propeller #1' through '#4', and fields for 'Propeller Serial No', 'Propeller TSI', 'Propeller TSO', and 'Propeller Total Time'; 'Event description' and 'Corrective action' (both with scrollable text areas); and three checkboxes at the bottom: 'Click to view table contents in XML format', 'Click to select and view a record in XML format', and 'Incident summary by aircraft Reg. No.'. A callout box on the left, titled 'Drop-down lists for existing records', has arrows pointing to the 'Maintenance Provider', 'Carrier Name', 'Model/Maker', and 'Part Number' fields. The status bar at the bottom shows 'Record: 1/1'.

III.3 Create a New SDR Report

The SDR form is to be filled the same way as MIS form

MIS SYSTEM - Developed by NSU under FAA grant

Action Edit Query Block Record Field Help Window

MIS
SDR
MAINTENANCE CO
PART
AIRCRAFT
ENGINE
PROPELLER
MAINTENANCE CO. USERS
AIRLINE USERS

SDR

U.S. Department of Transportation
 Federal Aviation Administration

DEPARTMENT OF TRANSPORTATION
 FEDERAL AVIATION ADMINISTRATION

FORM APPROVED
 OMB No. 2120-0008
RIS: WS 8070-1
 Control No.
 ATA Code

Service Difficulty Report
 AERONAUTICAL EQUIPMENT

MAJOR EQUIPMENT IDENTITY

Enter pertinent data

	MANUFACTURER	MODEL/SERIES	SERIAL NUMBER	N-
AIRCRAFT				
POWERPLANT				
PROPELLER				

PROBLEM DESCRIPTION

DATE	STATUS	CARRIER	ATA	AIRCRAFT TYPE	N-	CONTROL NO.

TEXT

SPECIFIC PART CAUSING PROBLEM

PART NAME	MFG. PART NUMBER	PART CONDITION	PART/DEFECT LOCATION

COMPONENT/APPLIANCE ABOVE PART INSTALLED ON

COMP/APPL NAME	MANUFACTURER	MFG. MODEL NUMBER	SERIAL NO.	Report whole hours	PART TT	PART TSO

SUBMITTED BY

SUBMITTER (Check one)

A	B	C	D	E	F	G	H	I	P.S.L.	ALERT	OPER/D.O.
CARRIER	REP	OPER	MCH	AIR	MFO	FAA	OTHER	Spec			
	ATA		TAAG								
<input checked="" type="radio"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>								

REC DOC NATURE STAGE STAT ROLL Frame CMC CMC

Record: 1/1

III.4 Generate Report on a Particular Aircraft

If you click this button:

The screenshot shows a web-based application window titled "MIS SYSTEM - Developed by NSU under FAA grant". The interface includes a menu bar with "Action", "Edit", "Query", "Block", "Record", "Eject", "Help", and "Window". Below the menu is a toolbar with various icons. The main content area is divided into several sections:

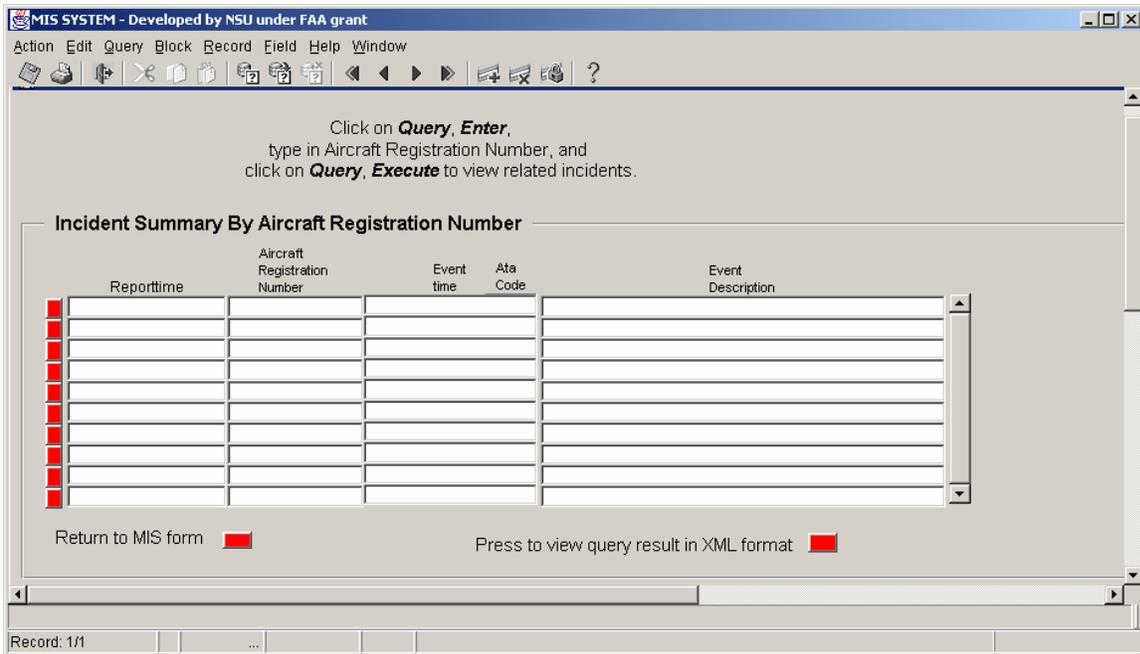
- MIS**: Record No. [text box], Report Time [text box], SDR No. [text box]
- SDR**: Maintenance Provider [dropdown], Carrier Name [dropdown], Event Time [text box]
- MAINTENANCECO**: Aircraft Reg No. [dropdown], Model/Maker [text box], Aircraft Serial No. [text box]
- PART**: Flight Related Data section with fields for Flight Num, Flight Hour, Depart Location, Flight Cycle, Depart Time, Sched, Depart Time, Actual, Destination, Sched, Destination, Actual, Arrival time, Sched, and Arrival time, Actual.
- AIRCRAFT**: ATA Code [text box]
- ENGINE**: Engine Related Data section with checkboxes for Engine #1, #2, #3, #4, Engine Serial No [dropdown], Engine TSI [text box], Engine TSO [text box], and Engine Total Time [text box].
- PROPELLER**: Propeller Related Data section with checkboxes for Propeller #1, #2, #3, #4, Propeller Serial No [dropdown], Propeller TSI [text box], Propeller TSO [text box], and Propeller Total Time [text box].
- MAINTENANCE CO. USERS**: Part Related Data section with fields for Part Number [dropdown], Manufacturer, Serial Number, Nomenclature, Part Vendor, and Date Installed.
- AIRLINE USERS**: Event description [text area] and Corrective action [text area].

At the bottom of the form, there are three buttons:

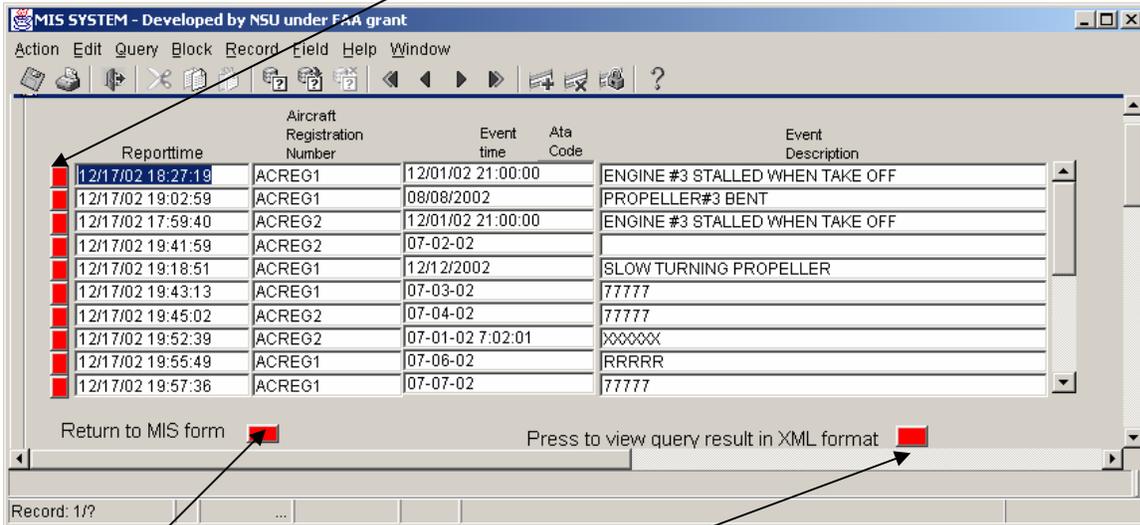
- Click to view table contents in XML format [checkbox]
- Click to select and view a record in XML format [checkbox]
- Incident summary by aircraft Reg. No. [button]

The status bar at the bottom left shows "Record: 1/1".

You will see:



Click Query -> Enter and then any Aircraft Reg number you wish to search or nothing (if all records are to be fetch). Now click Query -> execute, you will have a complete list of records. Then you can click any red button to the left to fetch that particular record



Click here is the user wishes to go back to MIS form.

If you click this button, you will see the record as an XML document

2. After you enter/tab (skip) over SDR No, you need to select or enter Maintenance Provider name. If the name is already in the drop-down list, you can just click to select and move on (tab) to the next field. If it is new after you type in the name, say FLIGHT-BY-NIGHT, and then press tab, you will see:

MIS SYSTEM - Developed by NSU under FAA grant

Action Edit Query Block Record E:ield Help Window

FAA

MIS

SDR

MAINTENANCECO

PART

AIRCRAFT

ENGINE

PROPELLER

MAINTENANCE CO. USERS

AIRLINE USERS

MAINTENANCE FACILITY

Name

Phone

Fax

Contact

E-mail

Click to view table contents in XML format

Click to select and view a record in XML format

Record: 1/1

After a form is filled, click Action -> Save to go back to the unfinished MIS form)

3. Next, select carrier name, and enter the Event Time. Whenever new database items need to be added, follow the same procedure as Maintenance Provider when entering new information on AIRCRAFT REG No, PART, ENGINE or PROPELLER
4. When you finished filling up the form click Action -> Save

IV. User Guide for Each Type of Users

IV.1 How to use SuperMISSDR.fmx

Special Notes to DBA

The Maintenance Co. User and Airline User blocks are used for inventory purpose only. User accounts must be first created by running the user.sql, grant.sql and the syn.sql before the username can be entered into the Maintenance User and Airline User tables. Company name or Carrier Name of the users must be entered by selecting from the existing list. Each update of an existing MIS report will be treated as an insertion of a new record into the MIS table.

Special caution must be taken when accessing records with this form to avoid accidental changing or deletion of data in the tables.)

1. Enter the following URL in the location box of your web browser (IE 5.0 or higher):
<http://servername:portnumber/servlet/f60servlet?form=SuperMISSDR3.fmx>
2. Enter the username, password and database name in the login window.
3. A new frame will be opened containing a form with multiple tabs on the left (see below sample) will be opened. Click on any of the tab to access a table in the database.

MIS SYSTEM - Developed by NSU under FAA grant

Action Edit Query Block Record Field Help Window

FAA

MIS Record No [] Report Time [] SDR No []

SDR Maintenance Provider [] Carrier Name [] Event Time []

MAINTENANCECO Aircraft Reg No [] Model/Maker [] Aircraft Serial No []

PART

AIRCRAFT

ENGINE

PROPELLER

MAINTENANCE CO. USERS

AIRLINE USERS

Flight Related Data

Flight Num [] Flight Hour []

Depart Location [] Flight Cycle []

Depart Time, Schd [] Depart Time, Actual []

Destination, Schd [] Destination, Actual []

Arrival time, Schd [] Arrival time, Actual []

ATA Code []

Part Related Data

Part Number []

Manufacturer []

Serial Number []

Nomenclature []

Part Vendor []

Date Installed []

Engine Related Data

Click in the box(s) next to the problematic engine(s) Engine #1 #2 #3 #4

Engine Serial No [] Engine TSI [] Engine TSO []

Engine Total Time []

Propeller Related Data

Click in the box(s) next to the problematic propeller(s) Propeller #1 #2 #3 #4

Propeller Serial No [] Propeller TSI [] Propeller TSO []

Propeller Total Time []

Event description []

Corrective action []

Click to view table contents in XML format Click to select and view a record in XML format Incident summary by aircraft Reg. No

Record: 1/1

4. Click on any of the tools in the top tool bar
 - a. to enter query (Query -> Enter or alternatively click on the ? icon),
 - b. to execute query (Query -> Execute or alternatively click the ? with an arrow icon),
 - c. to cancel a query (the ? with an X icon),
 - d. to insert a new record (the green plus sign icon),
 - e. to delete (the red x icon),
 - f. to browse previous or next record (the left or right pointing arrows),
 - g. to save a record after insertion or update (the diskette icon),
 - h. to print the current record (the printer icon)

The screenshot shows a window titled "MIS SYSTEM - Developed by NSU under FAA grant". The menu bar includes "Action", "Edit", "Query", "Block", "Record", "Field", "Help", and "Window". The toolbar contains icons for: a question mark (query), a question mark with an arrow (execute query), a question mark with an X (cancel query), a green plus sign (insert), a red X (delete), left and right arrows (navigation), a diskette (save), and a printer (print). The main form area is divided into several sections: "MIS" (Record No, Report Time, SDR No), "SDR" (Maintenance Provider, Carrier Name, Event Time), "MAINTENANCECO" (Aircraft Reg No, Model/Maker, Aircraft Serial No), "PART" (Flight Related Data: Flight Num, Flight Hour, Depart Location, Flight Cycle, Depart Time, Destination, Arrival time), "ENGINE" (ATA Code, Engine Related Data: Engine #1-4, Engine Serial No, TSI, TSO, Total Time), "PROPELLER" (Propeller Related Data: Propeller #1-4, Propeller Serial No, TSI, TSO, Total Time), "Event description", and "Corrective action". At the bottom, there are three blue buttons: "Click to view table contents in XML format", "Click to select and view a record in XML format", and "Incident summary by aircraft Reg. No.". The status bar at the very bottom shows "Record: 1/1".

- i. to view records in a table as XML file (blue button on bottom left),
- j. to view a particular record as XML file (middle blue button on the bottom),
- k. to view summary report by aircraft registration numbers (blue button on bottom right),
- l. to clear contents in the current block (Block -> Clear) on Menu bar,
- m. to clear contents in the current record (Record -> Clear) on Menu bar, or
- n. to exit from the form (Action -> Exit or alternatively click on the arrow out the door icon).

IV.2 How to use FAAInspector.fmx

Special Notes

Inspectors of FAA may not delete any records from the tables. They can insert new records into any of the tables through different blocks in the form by clicking on the respective tabs. Each update of an existing MIS report will be treated as an insertion of a new record into the MIS table.

1. Enter the following URL in the location box of your web browser (IE 5.0 or higher):
<http://servername:portnumber/servlet/f60servlet?form=FAAInspector.fmx>
2. Enter the username, password and database name in the login window.
3. A new frame will be opened containing a form with multiple tabs on the left (see below sample) will be opened. Click on any of the tab to access a table in the database.

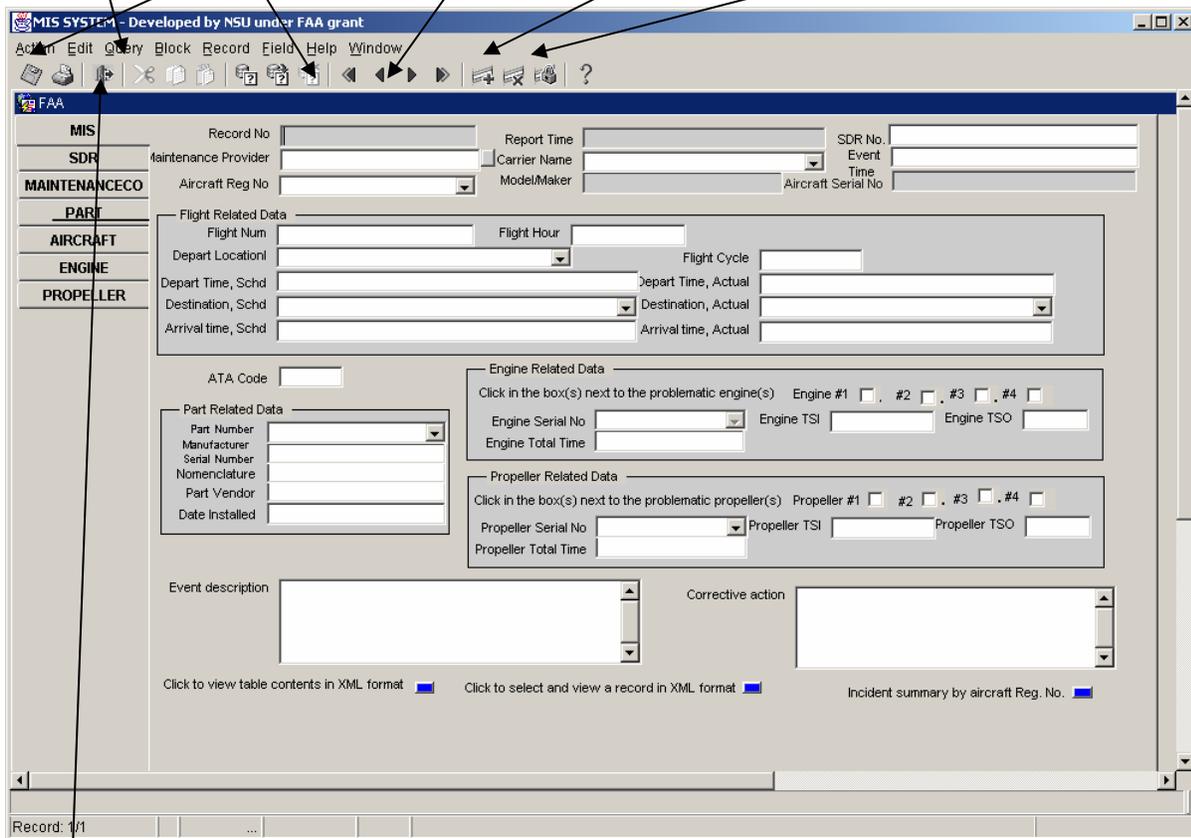
MIS SYSTEM - Developed by NSU under FAA grant

Action Edit Query Block Record Field Help Window

FAA

MIS Record No Report Time SDR No.
SDR Maintenance Provider Carrier Name Event Time
MAINTENANCECO Aircraft Reg No Model/Maker Aircraft Serial No
PART Flight Related Data
Flight Num Flight Hour
AIRCRAFT Depart Location Flight Cycle
ENGINE Depart Time, Schd Depart Time, Actual
PROPELLER Destination, Schd Destination, Actual
Arrival time, Schd Arrival time, Actual
ATA Code
Part Related Data
Part Number
Manufacturer
Serial Number
Nomenclature
Part Vendor
Date Installed
Engine Related Data
Click in the box(s) next to the problematic engine(s) Engine #1 #2 #3 #4
Engine Serial No Engine TSI Engine TSO
Engine Total Time
Propeller Related Data
Click in the box(s) next to the problematic propeller(s) Propeller #1 #2 #3 #4
Propeller Serial No Propeller TSI Propeller TSO
Propeller Total Time
Event description
Corrective action
Click to view table contents in XML format Click to select and view a record in XML format Incident summary by aircraft Reg. No.
Record: 1/1

4. Click on any of the tools in the top tool bar
 - o. to enter query (Query -> Enter or alternatively click on the ? icon),
 - p. to execute query (Query -> Execute or alternatively click the ? with an arrow icon),
 - q. to cancel a query (the ? with an X icon),
 - r. to insert a new record (the green plus sign icon),
 - s. to delete (the red x icon),
 - t. to browse previous or next record (the left or right pointing arrows),
 - u. to save a record after insertion or update (the diskette icon),
 - v. to print the current record (the printer icon)



- w. to view records in a table as XML file (blue button on bottom left),
- x. to view a particular record as XML file (middle blue button on the bottom),
- y. to view summary report by aircraft registration numbers (blue button on bottom right),
- z. to clear contents in the current block (Block -> Clear) on Menu bar,
- aa. to clear contents in the current record (Record -> Clear) on Menu bar, or
- bb. to exit from the form (Action -> Exit or alternatively click on the arrow out the door icon).

IV.3 How to use FAAAnlyst.fmx

Special Notes

Since this form is used by Analysts of FAA to query data for data analysis purposes only, therefore nothing in the table can be updated or deleted and no new records can be inserted.

1. Enter the following URL in the location box of your web browser (IE 5.0 or higher):
<http://servername:portnumber/servlet/f60servlet?form=FAAAnlyst.fmx>
2. Enter the username, password and database name in the login window.
3. A new frame will be opened containing a form with multiple tabs on the left (see below sample) will be opened. Click on any of the tab to access a table in the database.

MIS SYSTEM - Developed by NSU under FAA grant

Action Edit Query Block Record Field Help Window

FAA

MIS Record No [] Report Time [] SDR No []

SDR Maintenance Provider [] Carrier Name [] Event Time []

MAINTENANCECO Aircraft Reg No [] Model/Make [] Aircraft Serial No []

PART

AIRCRAFT

ENGINE

PROPELLER

Flight Related Data

Flight Num [] Flight Hour []

Depart Location [] Flight Cycle []

Depart Time, Schd [] Depart Time, Actual []

Destination, Schd [] Destination, Actual []

Arrival time, Schd [] Arrival time, Actual []

ATA Code []

Part Related Data

Part Number []

Manufacturer []

Serial Number []

Nomenclature []

Part Vendor []

Date Installed []

Engine Related Data

Click in the box(s) next to the problematic engine(s) Engine #1 #2 #3 #4

Engine Serial No [] Engine TSI [] Engine TSO []

Engine Total Time []

Propeller Related Data

Click in the box(s) next to the problematic propeller(s) Propeller #1 #2 #3 #4

Propeller Serial No [] Propeller TSI [] Propeller TSO []

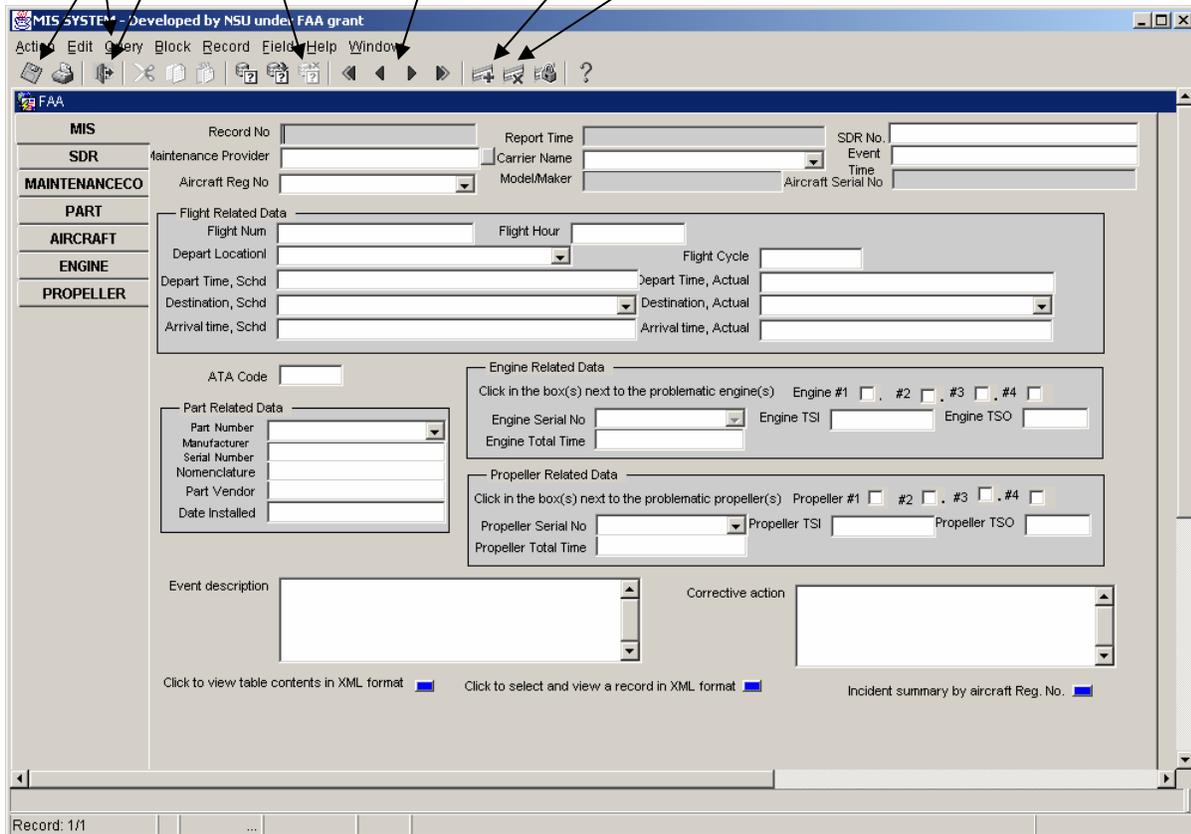
Propeller Total Time []

Event description [] Corrective action []

Click to view table contents in XML format Click to select and view a record in XML format Incident summary by aircraft Reg. No.

Record: 1/1

4. Click on any of the tools in the top tool bar
 - a. to enter query (Query -> Enter or alternatively click on the ? icon),
 - b. to execute query (Query -> Execute or alternatively click the ? with an arrow icon),
 - c. to cancel a query (the ? with an X icon),
 - d. to insert a new record (the green plus sign icon),
 - e. to delete (the red x icon),
 - f. to browse previous or next record (the left or right pointing arrows),
 - g. to save a record after insertion or update (the diskette icon),
 - h. to print the current record (the printer icon)
 - i. to view records in a table as XML file (blue button on bottom left),
 - j. to view a particular record as XML file (middle blue button on the bottom),
 - k. to view summary report by aircraft registration numbers (blue button on bottom right),
 - l. to clear contents in the current block (Block -> Clear) on Menu bar,
 - m. to clear contents in the current record (Record -> Clear) on Menu bar, or
 - n. to exit from the form (Action -> Exit or alternatively click on the arrow out the door icon).



IV.4 How to use MIS4MDepot.fmx

Special Notes

This form is mainly used by Maintenance Providers when filing MIS Report. The users of the form can insert new MIS records, or query existing MIS Reports filed by their own company. Any changes to an existing MIS report will be treated as an insertion of a new record. No records may be deleted from the tables through this form. The Maintenance Provider field in the MIS block will be pre-populated with the respective company name and can not be changed or erase from the report. The company may change information regarding its own company (except for the name of the company) through the MaintenanceCo Block by clicking on the **MAINTENANCECO Tab**

1. Enter the following URL in the location box of your web browser (IE 5.0 or higher):
<http://servername:portnumber/servlet/f60servlet?form=MIS4MDepot.fmx>
2. Enter the username, password and database name in the login window.
3. A new frame will be opened containing a form with multiple tabs on the left (see below sample) will be opened. Click on any of the tab to access a table in the database.

MIS SYSTEM - Developed by NSU under FAA grant

Action Edit Query Block Record Field Help Window

FAA

MIS Record No. Report Time SDR No.
SDR Maintenance Provider Carrier Name Event
MAINTENANCECO Aircraft Reg No. Model/Maker Aircraft Serial No. Time

PART
AIRCRAFT
ENGINE
PROPELLER

Flight Related Data
Flight Num Flight Hour
Depart Location Depart Time, Actual Flight Cycle
Destination, Sched Destination, Actual
Arrival time, Sched Arrival time, Actual

ATA Code

Part Related Data
Part Number
Manufacturer
Serial Number
Nomenclature
Part Vendor
Date Installed

Engine Related Data
Click in the box(s) next to the problematic engine(s) Engine #1 #2 #3 #4
Engine Serial No Engine TSI Engine TSO
Engine Total Time

Propeller Related Data
Click in the box(s) next to the problematic propeller(s) Propeller #1 #2 #3 #4
Propeller Serial No Propeller TSI Propeller TSO
Propeller Total Time

Event description Corrective action

Click to view table contents in XML format Click to select and view a record in XML format Incident summary by aircraft Reg. No.

Record: 1/1

4. Click on any of the tools in the top tool bar
 - a. to enter query (Query -> Enter or alternatively click on the ? icon),
 - b. to execute query (Query -> Execute or alternatively click the ? with an arrow icon),
 - c. to cancel a query (the ? with an X icon),
 - d. to insert a new record (the green plus sign icon),
 - e. to delete (the red x icon),
 - f. to browse previous or next record (the left or right pointing arrows),
 - g. to save a record after insertion or update (the diskette icon),
 - h. to print the current record (the printer icon)

- i. to view records in a table as XML file (blue button on bottom left),
- j. to view a particular record as XML file (middle blue button on the bottom),
- k. to view summary report by aircraft registration numbers (blue button on bottom right),
- l. to clear contents in the current block (Block -> Clear) on Menu bar,
- m. to clear contents in the current record (Record -> Clear) on Menu bar, or
- n. to exit from the form (Action -> Exit or alternatively click on the arrow out the door icon).

V. XML Documents

1. Purpose (A Quick introduction on XML)

When an electronic form is filled inside a web browser, the entered data can only be saved electronically as an HTML file. This saved format is not electronically reusable, since HTML files, are essentially coded in a page rendering language without any markup for document structure.

However, there is an open document exchange standard today that can replace the HTML without its data processing deficiencies: the XML (eXtensible Markup Language). XML is a powerful subset of SGML (Standard Generalized Markup Language, ISO 8879) tailored for Web deliveries. SGML and XML are being adopted as the open document exchange standard in various federal agencies as well as industrial and commercial enterprises. XML meets the business requirements that HTML failed on the following functionality:

- Data Validation – Data structural syntax can be enforced with XML parser, and data type validity can be checked.
- Data Reusability – Data hierarchical and structure are native to XML and they can be preserved without any loss of data intelligence.
- Data Interchangeability – XML standard is platform and application independent
- Data Longevity – XML, a meta-language, is open and extensible, not restricted to a single tag set like the HTML
- Data Security – The hierarchical structure in XML documents also intrinsically permits easy security enforcement with the digital signature specification for XML and component-wise data encryption. XML also permits finer grain of access control to render a document in different appearances for different users in full text or abridged form (with sensitive text hidden). This provides data confidentiality at no additional cost, since it would allow a document created once but published many times.

Therefore, we provide one-click buttons to return entered or retrieved form data into XML document(s) back to the user in a separate window. The end user can click File -> Save to save a local copy for electronic processing (stored in a local database or for business process integration) later.

Appendix B

Installation Guide

For the

FAA MIS/SDR Form Reporting System (Version 3.0)

Created by

**Ms. Deborah Chen
Department of Computer Science
Norfolk State University
Norfolk, VA 23504**

**Contact Person: Dr. Mou-Liang Kung
(757) 823-8650 (Voice)
mlkung@nsu.edu
(757) 823-9229 (fax)**

January, 2004

System and Software Configurations

1. Hardware and Software Requirements:

- a. Two Servers: PCs with Windows 2000 Professional or Server
 - **Oracle 9.0.1** and **Oracle Patch 9.0.1.4** should be installed on one PC to serve as the Oracle Database Server.
 - Install **Oracle 9iAS 1.0.2.2.2a** on the other PC first then, install **Oracle Form6i** and **Oracle Patch 12 for NT later** on the same machine as the Application/Web Server.
 - Both computers need to be configured and connected to the Internet.
- b. Clients: Computers with access to the Internet and the web browser IE 5.0 or higher.

2. Oracle 9i Database Server

- a. After installation of Oracle 9i, create a database according to following steps:
 1. Place the installation CD in your CD ROM/RW drive.
 2. If the ORACLE_HOME is in a drive other than C:\, open the file **FAAPART.bat** located in the **DATABASE** folder with a text editor and change all the "C:\" to the actual drive. Make the same modifications in the file **init.ora**.
 3. Copy the batch file **FAAPART.bat** into the **C:\Oracle\ADMIN\FAAPART\Scripts** folder (create the necessary folder(s) if not already exist). Run the .bat file from the **C:\Oracle\ADMIN\FAAPART\Scripts** folder and this will create a database instance called **FAAPART** and allocate tablespaces **PART_SPACE** and **PART_INDEX** for users.
- b. Create a user with **Resource** and **Connect** privileges by carrying out the following steps:
 1. Open a command window.
 2. Change the directory to your CD ROM/RW drive
 3. Change the directory to the **TABLESandUSERS** directory
 4. Enter the following command to create a new user by running the script file called **user.sql**:

```
sqlplus system/manager@FAAPART @USER XXXX *****  
PART_SPACE PART_INDEX
```

Where XXXX is the username and ***** is the password

- c. Create all user objects (tables, views, sequences and indexes etc.) for the project by entering the following command:

```
sqlplus XXXX/*****@FAAPART @FAATBLS
```

- d. Grant the new user with different privileges by first modify the available privileges in the *grant.sql* file. Then enter the following command to grant the new user with the desired privileges by running a script file call *grant.sql*.

```
sqlplus XXXX/***** @FAAPART @GRANT fauser
```

- e. Create other users repeat steps b. and d. first, then enter the following command to create synonyms for each new user:

```
sqlplus UUUUU/PPPPP@FAAPART @syn XXXX
```

Where UUUUU and PPPPP are the new user's username and password respectively and XXXX is the username of the user who owns the objects.

- f. Find the file *tnsnames.ora* in the Oracle Home /network/ADMIN folder and make sure the HOSTNAME and SID in the file are set according to the Oracle Server's IP address, the desired SID of your choice will be used to connect to the database in the future. In addition, also check to make sure the HOSTSERVER of the Oracle Listener file *listener.ora* is set properly so the applications installed on the application web server can have access to the database.

3. Oracle 9iAS Application/Web Server

- a. On the application web server machine, installed **Oracle 9iAS 1.0.2.2.2a** (choose *C:\iSuites* as its installation directory) and be sure to select and install **Oracle Form Server** in the process.
- b. Assuming that Oracle 9iAS is installed on the C:\ drive, create a folder called *My_XSQL* in the following directory

```
C:\ORACLE\iSuites\Apache\Apache\htdocs
```

- c. Open the *Htdocs* folder from the installation CD and copy the entire folder *My_XSQL* (disk copyies enclosed) into the following directory:

```
C:\ORACLE\iSuites\Apache\Apache\htdocs
```

- d. Stop the service **OracleiSuitesHTTPServer** by clicking Start -> Settings -> Control Panel -> Administrative Tools -> Services -> Right click on OracleiSuitesHTTPServer -> stop.
- e. **Copy the file *jserv.conf* from the installation CD (in the iASConf folder) into *C:\Oracle\iSuites\Apache\Jserv\conf*. Always keep a copy of the original *jerv.conf* file by changing its name (by appending the word ORIGINAL at the end of the file name) before copying. Modify the hostname according to the name of the computer.**
- f. **Copy the file *httpd.conf* (in the iASConf folder of the installation CD) to *C:\Oracle\iSuites\Apache\Apache\conf* folder. Modify the hostname from a text editor according to the name of the computer. Again, be sure to save a copy of the original *httpd.conf* before copying by changing its name (by appending the word ORIGINAL at the end of the file name) before copying.**
- g. Copy the file ***jserv.properties*** (located in the **iASConf** folder of the installation CD) into the ***C:\Oracle\iSuites\Apache\Jserv\conf*** folder. Don't forget to save a copy of the original *jerv.properties* file first by changing its name (by appending the word ORIGINAL at the end of the file name) before copying
- h. Open the file ***XSQLConfig.xml*** in the **iASConf** folder of the installation CD in a text editor and insert the actual username (currently mark as XXXX), password (currently mark as XXXX) of the user created in step 2 b above and the hostname of your Oracle server (currently mark as #####), then save it into the ***C:\Oracle\iSuites\xdk\lib*** folder. If there is a *XSQLConfig.xml* already in there, you may need to rename the original *XSQLconfig.xml* first in order to save the new one.

Security Note:

This file contains sensitive database username/password information that you want to keep secure on the server. This file can reside in any directory and it need only have read permission by the owner of the servlet engine under which the XSQLServlet is running.

On a PRODUCTION system, under no circumstances should this configuration file reside in a directory that can be browsed through the virtual path of your web server. Therefore, make sure that it is not in any directory or subdirectory of file system paths that are aliases to virtual directories in your web server environment.

- i. In the *C:\oracle\isuites\jdbc\lib* folder, make copies of the *class12.zip* and *class111.zip* files by changing their names (by appending the word ORIGINAL at the end of the file name), then copy the new *class12.zip* and *class111.zip* in the *C:\oracle\isuites\jdbc\lib* folder into the *C:\oracle\isuites\jdbc\lib* folder.
- j. Restart the server *OracleiSuitesHTTPServer C*

4. Installation of Oracle Form6i and Oracle Form Patch 12 (or higher) on the Application/Web Server. (Oracle Patches 12 (or higher) for NT can be downloaded from the ftp site: <ftp://oracle-ftp.oracle.com>)

- a. Install **Oracle Form6i Developer and Oracle Form Patch 12 (or higher) Configured to enable running forms over the web, be sure that “Oracle Jinitiator 1.1.8.19” is one of the components you select to install. Add the following lines to C:\Oracle\iSuites\network\ADMIN\tnsnames.ora**

```

Faapart =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS =
        (PROTOCOL = TCP)
        (Host = XXX.XXX.XXX.XXX)
        (Port = 1521)
      )
    )
    (CONNECT_DATA = (SID = FAAPART)
  )
)

```

- b. Add the following lines to **C:\Oracle\806\NET80\ADMIN\tnsnames.ora**

```

Faapart.world =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS =
        (PROTOCOL = TCP)
        (Host = XXX.XXX.XXX.XXX)
        (Port = 1521)
      )
    )
    (CONNECT_DATA = (SID = FAAPART)
  )
)

```

Note: the word “world” is the name of the actual default domain name.

- c. Assuming that the Oracle Home for Oracle Forms is C:\Oracle\806. Copy all the **.fmb** files from the **FORMS** folder of the installation CD into the **C:\Oracle\806\FORMS60** folder
- d. Open each of the **.fmb** files from Oracle Form Builder
- e. Connect to the database with the proper username, password and database name
- f. Compile the form using **File, Administration, Compile File** to create the executable **.fmx** file for each of the **.fmb** files.
- g. Close the Oracle FormBuilder.
- h. Open the file **formsweb.cfg** (in the **iASConf** folder of the installation CD) using a text editor and set the serverHost parameter value (currently is #####), port number (currently is 9000) and connection Mode (currently is HTTP) according to the hostname of your application web server then save it in the **Oracle\....\FORMS60\server** folder. Be sure the file type is still **.cfg** (not .txt). Be sure to keep a copy of the original formsweb.cfg for record by changing its name (by appending the word ORIGINAL at the end of the file name). You may need to stop the Oracle Form Server first in order to do this step.
- i. Copy the file **basejini.htm** (in the **iASConf** folder of the installation CD) into the **Oracle\....\FORMS60\server** folder. As usual, always keep a copy of the original file by changing its name (by appending the word ORIGINAL at the end of the file name) before copying.
- j. Stop and restart the **Oracle Forms Server [Forms60Server]** by clicking Start -> Settings -> Control Panel -> Administrative Tools -> Services -> Right click on Forms Server [Forms60Server] -> stop -> Right click on Forms Server [Forms60Server] -> start.

5. JInitiator

When a client calls any of the Oracle forms from the web browser for the first time; he/she will be prompt to install Jinitiator. User should just agree and accept the defaults of the installation when asked. This is a one-time-only process.

6. JDeveloper9.0.3

JDeveloper9.0.3 is installed for development of XML, XSL XSQL related applications as well as the platform for developing and deploying database webservises. A sample Webservice - Air_packWebService is provided with the deliverable CD. It needs to be deployed from JDeveloper9.0.3 on the Application Server after the OC4J standalone has been installed with the command:

```
JDEVELOPER_HOME\j2ee\home>java -jar oc4j.jar -install
```

and the OC4J container has been started with the following command:

```
JDEVELOPER_HOME\j2ee\home>java -jar oc4j.jar
```

7. Delivered Software and Document

- InstallationGuideV3.doc: this document
- UserGuide.doc: A brief introduction of different functionalities of various forms and their users' privileges.
- FAAPART.bat: a batch file to create database and to allocate tablespaces. And all the necessary script files and .ora files used by FAAPART.bat.
- user.sql, grant.sql and syn.sql: script files to create a user, and to grant privileges and create synonyms the new users.
- FAATbls.sql: the script file that creates all necessary objects as well as inserts all list values used in forms.
- Formsweb.cfg and basejini.htm: configuration files that set the parameters for running the forms from the web
- MIS4MDepot.fmb, FAAAnalyst.fmb, FAAInspector.fmb and SuperMISSDR3.fmb: Oracle form files.
- XSQLconfig.xml : The file contains database username/password information, and
- The My_XSQL folder containing all the necessary files to generate XML documents for data in the database.
- aircraftWebservice.jws – Jdeveloper9.0.3 workspace containing all files for the sample webservice.
- aircraft.Air_packWebServiceProxy.java : Java source code for sample webservice program
- aircraft.Air_packWebService.WSDL : Sample webservice description file

8. Special Notes

- a. Make sure the system variable path of the application web server machine begins with the following:

C:\ORACLE\iSuites\BIN;C:\Oracle\806\vbroker\bin;C:\ORACLE\806\bin

This will enable the starting of OracleiSuitesHTTPServer.

- b. Make sure the *OracleiSuitesHTTPServer* and *Oracle Forms Server [Forms60Server]* are started on the web server before running any of the .fmx over the web.
- c. The .fmb files must be recompiled if any changes have been made. Make sure no one is running the form by stopping the OracleFormsServer[Forms60Server] first, or the compilation will fail and no new .fmx file will be generated.

Appendix C

Developing a Standalone Web Service Client Accessing HAI Web Service

Step 1: e:\Microsoft.Net\FrameworkSDK\bin\wsdl /o:algo.cs
<http://65.111.185.28/algo/algo.wsdl>

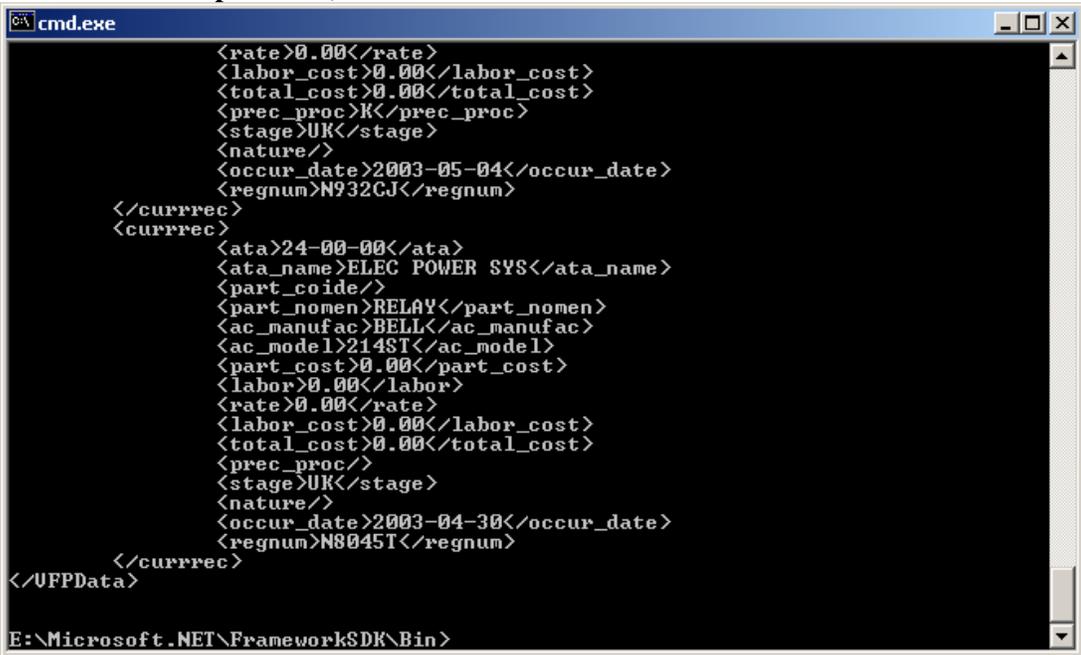
Step 2: csc /t:library algo.cs

Step 3: Create and compile TestMmir.cs:

```
public class TestMmir {  
  
public static void Main(System.String[] args){  
  
    algo mmir=new algo();  
    System.Console.WriteLine(mmir.GetMmirData("04222003", "04302003", "2"));  
  
}  
}
```

and `csc /r:algo.dll TestMmir.cs`

Step 4 Next, execution: `TestMmir` You should see



```
cmd.exe  
  
<rate>0.00</rate>  
<labor_cost>0.00</labor_cost>  
<total_cost>0.00</total_cost>  
<prec_proc>K</prec_proc>  
<stage>UK</stage>  
<nature/>  
<occur_date>2003-05-04</occur_date>  
<regnum>N932CJ</regnum>  
</currrec>  
<currrec>  
<ata>24-00-00</ata>  
<ata_name>ELEC POWER SYS</ata_name>  
<part_coide/>  
<part_nomen>RELAY</part_nomen>  
<ac_manufac>BELL</ac_manufac>  
<ac_model>214ST</ac_model>  
<part_cost>0.00</part_cost>  
<labor>0.00</labor>  
<rate>0.00</rate>  
<labor_cost>0.00</labor_cost>  
<total_cost>0.00</total_cost>  
<prec_proc/>  
<stage>UK</stage>  
<nature/>  
<occur_date>2003-04-30</occur_date>  
<regnum>N8045I</regnum>  
</currrec>  
</UFPData>  
  
E:\Microsoft.NET\FrameworkSDK\Bin>
```

APPENDIX D

(Student Paper Competition Winners)

Norfolk State University

First Place Winner

Dynamic Multi-Modal Multi-Platform
Web Publishing
at
Virginia Academy of Science
2003 Annual Meeting

By

Moses C. Allotey-Pappoe and Jacquelynn R. Dolly
Department of Computer Science
Norfolk State University
Norfolk, VA 23504
May 30, 2003

Norfolk State University

First Place Winner

Multi-Modal XML Publishing
at
National Technical Association
2003 Annual Meeting

By

Jacquelynn R. Dolly and Moses C. Allotey-Pappoe
Department of Computer Science
Norfolk State University
Norfolk, VA 23504
September 12, 2003

Appendix E

(Student paper)

Dynamic Multi-Modal Multi-Platform XML Publishing

By

Jacquelynn R. Dolly and Moses C. Allotey-Pappoe¹

Abstract

The demand to structure and describe data for the web and enterprise-based applications is ever-increasing due to the rapid development and adoption of XML technologies for Internet. Publishers are faced with the problems of restructuring, filtering, and redefining data into a format suitable for applications and different platforms, without rewriting the entire documents repeatedly. We propose to solve this problem by using two types of markup languages: XML and XSL. The task of writing and transforming XML applications into three formats, HTML, WML, and XHTML + VoiceXML will be developed as an application on the application (web) server, using many open-source products such as Apache Cocoon, Jakarta Tomcat, and Opera.

1. Introduction

XML is becoming a very powerful and useful language to use for structuring and describing data for the web. Based on the principle of create-once-publish-many, more and more data/documents are coded in platform neutral XML format to be used internally within a company or shared externally with partners (e.g. Business-to-Business web services) and customers (e.g. Business-to-Customer e-commerce). Because XML does not format its information for publishing and different businesses use their own formats and platforms, we need a way of easily transforming data from one format to another. We propose to solve this problem by using XSL Transformations or XSLT.

¹ **Acknowledgement:** The authors are currently supported by FAA Grant (Number 00-G-031) under the supervision of Dr. Mou-Liang Kung, Professor of Computer Science, Norfolk State University.

Disclaimer: The authors make no representation of any endorsement from FAA of this article. No part of this article should be interpreted as representing the official policies, either expressed or implied, of the FAA. FAA is not to be held responsible for the accuracy of the content in this Kit nor the views, opinions, and recommendations made therein.

In this document we will explain how we plan to use XML/XSLT to produce web pages into HTML, WML, and XHTML+VoiceXML (X+V) format. We will begin by giving a brief introduction to XML and XSL with sample documents. We will then describe how to use XSLT to produce three different formats, HTML, WML, and XHTML + VoiceXML. Finally we will proceed by explaining how to use our open-sources to produce a presentation format of your choice.

2. Introduction to XML

What is XML?

Extensible Markup Language (XML) is a markup language created by the World Wide Web Consortium (W3C). Martin Bryan, author of “An Introduction to the Extensible Markup Language (XML)”, also define it to be a “... subset of the Standard Generalized Markup Language (SGML) defined in ISO standard 8879:1986 that is designed to make it easy to interchange structured documents over the Internet.” ([1]) According to Nazmul Idris, author of “Benefits of Using XML”, XML is platform-independent, meaning, “...programs running on disparate platforms can communicate with each other.”([5]) XML offers a flexible and convenient method for representing encapsulated data. It uses information modeling rather than traditional data modeling, which allows us to create information, which can be consumed by human as well as machines. Unlike HTML with predefined tags meaning, XML allows the author of the document to define their own tags. This makes it a very flexible way for the author to describe what kind of data he/she has.

A typical XML document starts with the declaration of the XML header. It is then followed by the starting tag of a unique root element. Following this root element are other elements called the child element and within these child elements, we can have well nested sub elements that become a sub child element to the root. The end tags of the root element always end the XML document. The following is a segment of an XML document (See Appendix B for the full XML document):

```
<CS>
  <Head>
    <Name>Mou-Liang Kung</Name>
    <Title>Department Head</Title>
    <Phone>(757) 823-9454</Phone>
  </Head>
</CS>
```

From the above example, `<CS>` defines our XML root element, `<Head>` becomes our child element, and every tag seen within the child element defines our sub child element.

Types of XML Documents

Based on the syntax for writing or creating an XML document, XML documents can be put into two groups:

- **Valid Documents**
These are XML documents that conform to rules defined by DTD or XSchema. Typically, they can be validated by an XML parser.
- **Well-Formed Documents**
These XML documents do conform to the XML syntax rules and don't conform to any predefined DTDs or XSchema..

3. Introduction to XSL

What is XSL?

Although XML is good for structuring and describing data, it does not describe the presentation of that data at the same time. Extensible Stylesheet Language (XSL) is a language used to describe how an XML document should be transformed into an HTML, WML, Portable Document Format (PDF), VoiceXML, etc. XSL documents typically have .xsl extension in their file names.

Components of an XSL

According to W3Schools, there are three XSL related standards:

- **XSL Formatting Objects (XSL-FO)**
This is a language for formatting XML documents. This is done by defining a set of elements which in turn helps describe how an XML document should be formatted. This style sheet is normally used to generate high-quality printable documents.
- **XML Path Language (XPath)**
This is the language used to define locations or parts in an XML document. This therefore enables us to define which portion to transform on our XML document.
- **XSL Transformations (XSLT)**
This an XML vocabulary used for transforming XML documents into different formats. It is part of the XSL that makes the decision on which elements to allow the web browser to display. ([8])

A. Sample XSL Stylesheet

An XSL stylesheet is an XML document and, therefore, always starts with the declaration of the XML version used on the first line. It is then followed by the declaration of the style sheet version used. This part of the document serves as the root element and also enables one to tell that, the document is an XSL style sheet document. The rest of the code will depend on which format we want to transform our XML document to. Below is a segment of a style sheet that transforms an XML document sample (See Appendix B) into an HTML format in a table form:

```
<xsl:template match= "/">
  <html><body>
    <h2> NSU Computer Science Department</h2>
    <xsl:value-of select = "CS/Head /Name" />
    <xsl:value-of select = "CS/Head /Title" />
    <xsl:value-of select = "CS/Head /Phone" />
  </body></html>
</xsl:template>
```

- How an XSL transforms an XML document

When an XSL stylesheet is applied to an XML document, a transformation process takes place in order to produce the presentation of that XML source content that was intended by the designer of the stylesheet. First, XSLT will have to locate all elements of the XML document that have to be transformed. This is achieved by using the definitions defined by XPath in the style sheet.

In the above XSL document sample, we use `<xsl:template match= "/">` to let the XSLT know that, any match made from `"CS/Head/Name"`, `"Title"`, or `"Phone"` will be transformed. Therefore when a match is made, XSLT takes that portions of the document and transforms it into the format specified by the style sheet. Since our style sheet defines an HTML format, XSLT will transform all matches made into HTML for output.

4. Dynamic XML Publishing

What does "Multi-Modal Mean?"

Dynamic multi-modal multi-platform XML publishing is simply the dynamic publishing of the same XML documents into multi-modal applications that can be viewed on different platforms.

In a publication posted on the internet by IBM Alphaworks, a multi-modal application was defined to be an application that "...offers end users a flexible interface that allows interaction

with their device using a combination of keyboard, touch screen, stylus, telephone keys, and voice.”([4])

What is WML?

Wireless Markup Language (WML) is an XML-based meta language used with wireless devices, such as mobile phones, PDAs, handhelds, and so on. Just like XML, WML is a subset of HTML but has its roots in XML. Instead of being based on web pages, WML has a basic unit called a deck which is composed of one or many cards. A deck is simply a wireless site and a card is the site’s page. Once the deck of cards is downloaded, the user is shown a card in which he/she can view it and then move between other preloaded cards in the deck. The following is a segment from an XSL document which has a deck with one card that shows the computer science’s department head information:

```
<xsl:template match="/">
  <wml>
    <card id = "department head" title = "Computer Science Department">
      <p>
        <xsl:for-each select="CS/Head">
          <xsl:value-of select="Name"/><br/>
          <xsl:value-of select="Title"/><br/>
          <xsl:value-of select="Phone"/><br/>
        </xsl:for-each>
      </p>
    </card>
  </wml>
</xsl:template>
```

As you can see, the basic structure of the language is similar to XML. The body of the WML document is enclosed within <wml>...</wml> tags. A card element starts with a <card> tag and ends with a </card> tag. It contains the title attribute, and the ID attribute, used to navigate between cards.

What is X+V?

X+V is a multi-modal markup language standard that Motorola, Opera Software ASA, and IBM submitted as a proposal to the World Wide Web Consortium (W3C). This markup language is used in the creation of multi-modal applications and it’s the combination of Extensible HyperText Markup Language (XHTML), a visual component, and VoiceXML, a voice component. These two components, visual and voice, are linked together using XML Event Handlers: “event”, “handler”, and “listener”.

The purpose of the event is to handle focus given to a field. This causes the handler to be executed when the cursor gives focus, or enters a particular field specified by the handler. The handler serves as a connector for the VoiceXML element and the HTML element. The

VoiceXML form “id” is what the handler uses to connect the VoiceXML element to the HTML element. The listener part of the XML Event Handler enables the execution of the handler whenever an event occurs.

Atypical X+V file has .xml or .html as it’s file extension. Also, the first three lines in this file is declared with the namespace for VoiceXML and the XML Events Handler as follows:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C/DTD XHTML+Voice 1.0/EN" "xhtml+voice.dtd">
<html xmlns=http://www.w3.org/1999/xhtml
      xmlns:ev=http://www.w3.org/2001/xml-events
      xmlns:vxml="http://www.w3.org/2001/vxml"
      xml:lang="en_US">
```

Following this declaration is the definition of the VoiceXML elements. These elements are included within the <head> tags of the document. The rest of the document contains the definition of the XHTML elements with their XML Event Handlers. These are defined in the <body> tags of the document.

What is VoiceXML?

Voice Extensible Markup Language (VoiceXML) is an XML-based language “...designed for creating audio dialogs that feature synthesized speech, digitized audio, recognition of spoken and DTMF key input, recording of spoken input, telephony, and mixed initiative conversations.” ([2]) The VoiceXML forum, founded by AT&T, IBM, Lucent Technologies, and Motorola, developed VoiceXML so that it could give authors control over voice dialogs between the users the application. It uses a voice browser’s text-to-speech system to synthesize speech and recordings for audio output. The user’s input is converted into textual input using powerful Automatic Speech Recognition (ASR) software along with a server that is connected to the Internet.

Components of a VoiceXML document

All VoiceXML documents begin and end with the <vxml> tag, which is mainly a container for dialogs. The two types of dialogs are forms and menus. The <form> element is the key element in a VoiceXML document; every document should have at least one <form> element. Forms are primarily for presenting information and gathering input. The main components of a form are: <field>, <prompt>, <grammar>, and <block>. The <prompt> element uses the TTS text to prompt the user and the <field> element is used to gather the users input. The <grammar> element is used to specify a grammar recognition grammar. Menus are used to specify what to do next.

The following is a segment of a VXML document that welcomes the user to the home page, asks him/her for a choice of emphasis, and then goes to the user’s choice:

```

<vxml:form id="main_menu">
  <vxml:field name="user">
    <vxml:prompt>Welcome to Norfolk State University, Department of Computer
    Science. For information on Computer Science, General Program, say general
    program. For Computer Engineering, say engineering.
    For Information Systems, say i s. For Computer Science, de knee mus General
    Option, say general option. For Computer Science, de knee mus Engineering Option,
    say engineering option. to quit, say exit.</vxml:prompt>
  <vxml:filled>
    <vxml:assign name="user_input" expr="user"/>
  </vxml:filled>
</vxml:field>
</vxml:form>

```

Advantages of VoiceXML

There are many advantages that VoiceXML can bring to the world. Wireless interfaces with WAP/XHTML browsers have the disadvantages of having small screens and keypad input which makes it harder to use, especially while driving. VoiceXML can open up huge opportunities. The telephone is one of the most powerful technologies used today. According to the VoiceXML Forum, "...the phone is important. There are over 1.5 billion phones in use, far more than there are Internet-connected computers... Telephone networks are much more reliable than data networks." ([7]) Voice applications will help increase revenue in the mobile phone industry. More people will purchase wireless phones, increasing the revenue for mobile providers. Users will spend more time on their wireless phones, therefore increasing not only the number of minutes used but also the revenue for mobile service providers. Voice applications will also help users, especially in cases like driving where pressing the keypad all of the time can be very dangerous. VoiceXML will also be important for the visually impaired. Not only will they be able to access the Internet with ease, it will also help them with things like the radio, the TV (for purposes like the news), and even answering the phone. They could have a remote that has a push-to-talk button to control whichever appliance they are currently using.

5. Implementation

This section briefly describes how Apache Cocoon, working along with Jakarta Tomcat, will be used to dynamically publish multi-modal applications and web pages onto different platforms. We will use the Opera web browser with IBM's multi-modal browser plug-in to display our dynamic applications: HTML, WML, and XHTML + VoiceXML. We will begin by briefly describing what Apache Cocoon, Jakarta Tomcat, and the Opera web browser is. We will then proceed to the configuration of our web server to enable the dynamic publishing of our applications.

Apache Cocoon is a servlet that serves as a powerful publishing framework for publishing XML and XSL documents into different formats like HTML, WML, PDA, etc and onto different device. Apache Cocoon needs a servlet engine for it to able to run. Since we will be doing dynamic web publishing, we will need a servlet engine that can publish its web documents dynamically. This therefore calls for the use of Jakarta Tomcat as our servlet engine (See Appendix D for the installation and configuration of Apache Cocoon and Jakarta Tomcat).

The Opera web browser is a browser that supports multiple operating systems. Aaron Turpen, a Web professional and author of “Web Browser Choices”, described it to be “...the fastest loading and probably the most compliant...” of all the popular browsers. ([6]) The Opera browser also supports the viewing of web standards: HTML, WML, XML, XHTML, CSS, and DOM. With the installation of a multi-modal browser plug-in on Opera, Opera can be used to view multi-modal applications.

Dynamic Publishing to a HTML Format

The first step in producing a dynamic HTML format is to create an XML document and an XSL stylesheet to transform the XML document into HTML. The XML document should be saved under the following directory: TOMCAT_HOME// Webapps/ Cocoon/ docs/ samples/. In creating the stylesheet, we will have to declare the document to be an XSL style sheet. The correct way to declare an XSL style sheet according to the W3C XSL Recommendation is:

```
<xsl:stylesheet version="1.0"xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
```

Now we have to configure our web server to generate an HTML document. Edit the pipeline portion of the sitemap.xml file located in the TOMCAT_HOME//Webapps/Cocoon by adding the following lines to it:

```
<map:match pattern="nsu.html">  
  <map:generate src="docs/samples/nsu.xml"/>  
  <map:transform src="stylesheets/page/nsuHTML.xsl"/>  
  <map:serialize type="html"/>  
</map:match>
```

Note: nsu.html is the name for the web document generated dynamically, nsu.xml is the XML document and nsuHTML.xsl is the XSL (stylesheet) document

Start the Tomcat Web server, open the Opera web browser, and request for this URL:
<http://localhost:8080/cocoon/nsu.html>

You should see something similar to the following:

Dynamic Publishing to a WML Format

The first step in producing a dynamic WML format is to create an XML document and an XSL stylesheet to transform the XML document into HTML. In our case we will use the same XML document as used for the HTML format. Because an WML document is no more than an XML document, so it has to include the XML declaration - defines the XML version and the character encoding used in the document. The DOCTYPE is defined to be wml, and the DTD is accessed at www.wapforum.org/DTD/wml_1.1.xml. This will make the heading of our stylesheet look like this:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
```

To configure the web server to generate a WML document, the sitemap should be edited by adding the following:

```
<map:match pattern="nsu.wml">
  <map:generate src="docs/samples/nsu.xml"/>
  <map:transform src="stylesheets/page/nsuWML.xsl"/>
  <map:serialize type="wml"/>
</map:match>
```

Note: nsu.wml is the name for the web document generated dynamically, nsu.xml is the XML document and nsuWML.xsl is the XSL (stylesheet) document

Start the Tomcat Web server, open the Opera web browser, and request for this URL:
<http://localhost:8080/cocoon/nsu.wml>

You should see something similar to the following:

Dynamic Publishing to a Multi-Modal Format (X+V)

Using the same XML document as above we now have to create a stylesheet to transform the XML document into a multi-modal application. In creating the stylesheet, we will have to add the declaration of the namespace for VoiceXML and the XML Event Handlers to the declaration of version of the stylesheet being used. This will make the heading of our stylesheet look like this:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:ev="http://www.w3.org/2001/xml-events"
  xmlns:vxml="http://www.w3.org/2001/vxml" xml:lang="en_US">
```

After the heading declaration, we have to define the VoiceXML elements needed. These are included in the <head> tag of your document. Following the <head> tag is the <body> tag. Define all your XHTML elements with their respective XML Event Handlers. Save this document to TOMCAT_HOME//Webapps/Cocoon/stylesheets/page/ directory.

Now we have to configure our web server to generate an HTML document or MXML document. We will use the configuration for the generation of an HTML document in this case. Edit the pipeline portion of the sitemap.xmap file located in the TOMCAT_HOME//Webapps/Cocoon by adding the following lines to it:

```
<map:match pattern="nsuVoice.html">
  <map:generate src="docs/samples/nsu.xml"/>
  <map:transform src="stylesheets/page/nsuVOICE.xsl"/>
  <map:serialize type="html"/>
</map:match>
```

Note: nsuVoice.html is the name for the web document generated dynamically, nsu.xml is the XML document and nsuVOICE.xsl is the XSL (stylesheet) document

Start the Tomcat Web server, open the Opera web browser, and request for this URL:
<http://localhost:8080/cocoon/nsuVoice.html>

You should see the visual of the multi-modal application depending on your stylesheet. Click in the appropriate fields to hear the voice part the application. To enter any input by voice, hold down the Scroll Lock on the keyboard, and say your word.

Difficulties Encountered

After from writing a style sheet to transform the XML document to look the way we it want to, we barely had any difficulties with the dynamic generation of our HTML and WML document.

With our multi-modal application, we had troubles with the declaration of the style sheet version with the other headings needed to produce this application. We finally got to solve this by just declaring the XML version and adding namespace for the VoiceXML elements and the XML Event Handler to the declaration of the style sheet version. This therefore made our header declaration look like this:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:ev="http://www.w3.org/2001/xml-events"
  xmlns:vxml="http://www.w3.org/2001/vxml"
  xml:lang="en_US">
```

On the same application, we found it difficult to go from one VoiceXML element to another using the `<vxml:goto next= " " >` element. After a better understanding of how the XML Event Handler works, we were to manipulate our VoiceXML elements.

Also, since our web document is generated dynamically, it was difficult for us to create a Java Speech Grammar Format (JSGF) file and add it to our voice form field. We had to fine all the valid inputs using the in-line grammar definition. This is made our grammar part look like this:

How Apache Cocoon connects to a Database

Depending on which database we are accessing, JDBC and XML:DB API are to be selected to connect to MySQL and Xindice respectively.

Connecting to MySQL

According to MySQL Reference Manual on the Internet, MySQL is an open source server that serves as a relational database management system. This enables the storage of data in a tabular manner. Due to its structured way of collecting data, MySQL adds speed, security and flexibility to its data access.

Using JDBC drivers, a programming-level interface for communicating with databases in a uniform manner, Cocoon can be connected to MySQL. But first, we have to download and install MySQL and the JDBC drivers (See Appendix F). Now, use the steps below to complete the connection process:

- Edit the load classes of web.xml located in TOMCAT_HOME/Webapps/Cocoon/WEB-INF directory to look like this:

```
<init-param>
<param-name> load - class </param-name>
  <param-value>
    .
    .
    <!-- For MySQL -->
    com.mysql.jdbc.Driver
  </param-value>
</init-param>
```

- Create a simple database named *Dates* with *NSUDates* being its tables and *month*, *day* and *year* being its columns and insert some dates into it. See MySQL website (mysql.com) on how to create this database.
- Edit the *cocoon.xconf* file located in TOMCAT_HOME/Webapps/Cocoon/WEB-INF directory to look like this:

```

<datasources>
  <jdbc name= "test">
    <dburl> jdbc:mysql://localhost:3306/Dates </dburl>
    <user> root </user>
    <password></password>
  </jdbc>
</datasources>

```

This will establish a connection to our database. *localhost* can be replaced with our host name, *3306* can be replaced with our MySQL port, *nsudates* can be replaced with our database name, *root* can be replaced with the user name used to connect to our database and if a password is required, we can also include that in the password portion.

- Use the sample below to create an XML document and save as *sample.xml*

```

<?xml version= "1.0"?>
<example>
  <sql:execute-query smlns:sql=
    "http://apache.org/cocoon/SQL/2.0">
    <sql:use-connection> test </sql:use-connection>
    <sql:query>
      select month, day, year from nsudates
    </sql:query>
  </sql:execute-query>
</example>

```

- Now we need a stylesheet to retrieve all the information needed. Use the example and save the document as *sample.xsl*

```

<?xml version= "1.0"?>
<xsl:stylesheet version= "1.0" xmlns:xsl=
http://www.w3.org/1999/XSL/Transform
xmlns:sql= "http://apache.org/cocoon/SQL/2.0">

<xsl:template match= "example">
  <html>
    <body><table>
      <xsl:apply - templates select= "sql:rowset/sql:row" />
    </body></table>
  </html>
</xsl:template>

<xsl:template match= "sql:row">
  <tr> <xsl:apply - templates /> </tr>
</xsl:template>

<xsl:template match= "sql:month | sql:day | sql:year">
  <td> <xsl:value - of select = "." /> </td>
</xsl:template>

</xsl:stylesheet>

```

- Modify the sitemap.xmap file located in the TOMCAT_HOME/Webapps/Cocoon directory by adding the code below to the pipeline portion:

```

<map:pipeline>
  <map:match pattern = "sampleSQL">
    <map:generate src = "sample.xml" />
    <map:transform type = "sql" />
    <map:transform src = "sample.xsl" />
    <map:serialize />
  </map:match>
</map:pipeline>

```

- Start MySQL server.
- Open an Internet browser and enter <http://localhost:8080/cocoon/sampleSQL>
- If the previous steps were followed correctly, the user should see the data stored in the database.

Now that we have connected Cocoon to MySQL, let's see how it is done with Xindice.

Connecting to Xindice

Xindice, pronounced zeen-dee-chay, is an open source native XML database management system written in java and maintained by the Apache organization. It is designed to ease the storing of XML data. At the present time Xindice uses XPath for its query language and XUpdate for its update language. Additionally, it also implements the XML:DB API, which is similar to ODBC and JDBC, but has different levels of interoperability. Xindice stores all of its XML documents inside of collections. “While collections can be used strictly for organizational purposes, Xindice also allows for indexes to be created on collections to increase XPath performance” (Liotta). Xindice comes with a set of command line tools (Appendix H) for using the database. In order to set up and use Xindice see Appendix G.

- Go to `sitemap.xmap` and look for and delete the following code snippet:

```
<map:generator name="xmldb"
  src="org.apache.cocoon.generation.XMLDBGenerator"
  logger="sitemap.generator.xmldb">
  <driver>org.dbxml.client.xmldb.DatabaseImpl</driver>
  <base>xmldb:dbxml:///db</base>
</map:generator>
<map:generator name="xmldbcollection"
  src="org.apache.cocoon.generation.XMLDBCollectionGenerator"
  logger="sitemap.generator.xmldbcollection">
  <driver>org.dbxml.client.xmldb.DatabaseImpl</driver>
  <base>xmldb:dbxml:///db</base>
</map:generator>
```

and

```
<map:match pattern="xmldb-generator/db/**/">
  <map:generate type="xmldbcollection" src="{1}"/>
  <map:serialize type="xml"/>
</map:match>
<map:match pattern="xmldb-generator/db/**">
  <map:generate type="xmldb" src="{1}"/>
  <map:serialize type="xml"/>
</map:match>
```

- Change the part in `sitemap.xmap` that reads
`xmldb:dbxml://localhost:4080/...` to
`xmldb:xindice://localhost:4080/db/...` so that you end up with:

```

<map:match pattern="xmldb/**">
  <map:match type="request-parameter" pattern="xpath">
    <map:generate
      src="xmldb:xindice://localhost:4080/db/{../1}#{1}"/>
    <map:serialize type="xml"/>
  </map:match>
  <map:generate src="xmldb:xindice://localhost:4080/db/{1}"/>
  <map:serialize type="xml"/>
</map:match>

```

- Look for the following code snippet in `cocoon.conf`: `<driver type="dbxml" class="org.dbxml.client.xmldb.DatabaseImpl"/>` and change it to read: `<driver type="xindice" class="org.apache.xindice.client.xmldb.DatabaseImpl"/>` you should end up with:

```

<source-handler logger="core.source-handler">
  <!-- xmldb pseudo protocol -->
  <protocol name="xmldb"
    class="org.apache.cocoon.components.source.XMLDBSourceFactory">
    <!-- Xindice driver -->
    <driver type="xindice"
      class="org.apache.xindice.client.xmldb.DatabaseImpl"/>
    <!-- Add here other XML:DB compliant databases drivers -->
  </protocol>
</source-handler>

```

- Copy `xindice.jar` from: `%XINDICE_HOME/java/lib` to: `cocoon\WEB-INF\lib`
- Start Xindice, if it isn't running.
- Start Tomcat
- If you followed the install instructions completely you should have address book example collection already created. To test Xindice, try:
`http://localhost:8080/cocoon/xmldb/addressbook/` and you should end up with:

```

<?xml version="1.0" encoding="UTF-8"?>
<collection:collections resources="3" collections="0"
  xmlns:collection="http://apache.org/cocoon/xmldb/1.0">
  <collection:resource name="0100007f1f9fdabb000000ec2b2a4110"/>
  <collection:resource name="0100007f1f9fdabb000000ec2b30c130"/>
  <collection:resource name="0100007f1f9fdabb000000ec2b8cabd3"/>
</collection:collections>

```

6. Conclusion

This research experience has taught us many things. We have learned a lot about XML and its many advantages, especially to those in the business industry. We have also learned how to use Apache Cocoon and Jakarta Tomcat to dynamically publish XML documents into several formats and onto different platforms (e.g. pda, and audio). This research has benefited us and can benefit others also. We are now able to offer high-paying businesses flexible and convenient ways of presenting encapsulated data without breaking the contributions of others.

At this point of our research, we are in the process of building a 3-tier client/server dynamic publishing system in order to dynamically store and retrieve XML documents from two types of databases: XML-enabled relational databases, and native XML databases. We are going to be using many open-source products such as Apache Cocoon, Jakarta Tomcat, MySQL, and Xindice.

Bibliography

1. Bryan, Martin. "An Introduction to the Extensible Markup Language (XML)," <http://www.personal.u-net.com/~sgml/xmlintro.htm>, The SGML Centre, 1997.
2. Chamberlin, Don, et al; "<http://www.w3.org/> W3C Working Draft, Nov 2002.
3. Cagle Kurt Cagle, et al; "Professional XSL" June 2001, Birmingham, UK: Wrox Press
4. IBM Alphaworks, "Multimodal Browser and Toolkit," <http://www.alphaworks.ibm.com/aw.nsf/FAQs/mmb>, Jan 2003
5. Idris, Nazmul "Benefits of Using XML," <http://developerlife.com/xmlbenefits/default.htm#93332>, Jun 1999.
6. Turpen, Aaron "Web Browser Choices," <http://www.allwebarticles.com/articles/pfv.php>
7. VoiceXML Forum, "Voice eXtensible Markup Language (VoiceXML™) version 1.0," <http://www.w3.org/TR/voicexml/>
8. W3Schools, <http://www.w3schools.com>

Appendix A

XML Syntax Rules for Well-Formed XML Documents:

- must begin with an XML declaration
- must have one unique root element
- all start tags must match end-tags
- XML tags are case sensitive
- all elements must be closed
- all elements must be properly nested
- all attribute values must be quoted
- XML entities must be used for special characters

Source taken from the W3Schools website
(http://www.w3schools.com/schema/schema_why.asp)

Appendix B

Sample XML Document:

---- *NSU.xml* ----

```
<?xml version='1.0'?>
  <CS>
    <CS-Title>Computer Science</CS-Title>
    <Head>
      <Name>Mou-Liang Kung</Name>a
      <Title>Department Head</Title>
      <Phone>(757) 823-9454</Phone>
    </Head>
    <About>
      <p>The Computer Science Program is designed to provide students with
      fundamental training in the theoretical and practical aspects of computer
      science. Coupled with the program's strong mathematics component, this
      training provides graduates with the necessary background for employment in a
      wide variety
      of computing fields or for a smooth entry into graduate level study.</p>

      <p>In addition, the Department provides service courses to assist students of all
      majors in mastering fundamental computer concepts and a variety of
      programming languages.</p>
```

<p>The Department of Computer Science offers the B.S. Degree in Computer Science which includes the emphasis areas:</p>

<Lists>

- <list1>-Computer Science (General Program) </list1>
- <list2>-Computer Engineering </list2>
- <list3>-Information Systems </list3>
- <list4>-Computer Science (DNIMAS- General Option) </list4>
- <list5>-Computer Science (DNIMAS- Engineering Option) </list5>

</Lists>

<p>The program addresses a number of career opportunities within the curriculum. The Computer Engineering option is suitable for students who are interested in the design and implementation of hardware. The Information Systems option qualifies students for employment in business environments.</p>

</p>

</About>

<GeneralRequirements>

<RequirementsTitle>GENERAL DEPARTMENT REQUIREMENTS</RequirementsTitle>

<p>Computer Science majors must complete 120 credits to complete the B.S. degree. Additionally:</p>

<Lists>

- <list1>Students must meet prerequisites or their equivalents prior to enrolling in more advanced computer science courses.</list1>
- <list2>Computer science majors must earn at least a "C" grade in all science, mathematics and computer science courses. </list2>
- <list3>Computer science majors are not permitted to enroll in any course whose content is prerequisite, or assumed knowledge, for a higher level course that a student has already completed. </list3>
- <list4>Each computer science major must pass a comprehensive examination covering the content of the core courses through CSC 390.</list4>

</Lists>

<p>The B.S. degree program in Computer Science (General, Computer Engineering, and DNIMAS Options) is "accredited by the Computer Science Accreditation Commission (CSAC) of the Computing Sciences Accreditation Board (CSAB), a specialized accrediting body recognized by the Council on Post-secondary Accreditation (COPA) and the U.S. Department of Education."</p>

</GeneralRequirements>

<ComputingFacilities>

<CF-Title>COMPUTING FACILITIES</CF-Title>

<Micro-Lab>

<Title>Microcomputer Laboratory</Title>

<p>This laboratory is located in Room C-224 Brown Hall and contains 49 (IBM compatible and Macintosh) microcomputers. Microcomputer instructional software includes spreadsheets, compilers/Interpreters (Visual BASIC, Visual C++, Prolog, Pascal, Smalltalk, and XLISP), windows and graphics software, database packages, word processors, expert systems, web browser, and utilities.</p>

</Micro-Lab>

<Unix-Lab>

<Title>Microcomputer and UNIX Laboratory</Title>

This Laboratory is located in Room C-231 Brown Hall and contains 15 SUN workstations and 27 pentium multimedia computers. Software available includes CASE Tools, DesignCAD, MathCAD, Oracle, Simfactory, Transputers, Presentation applications, symbolic algebra software (Mathematica) and the software cited for the Microcomputer Laboratory. </p>

</Unix-Lab>

</ComputingFacilities>

<ElectrncClassroom>

<p>The Electronic Classroom in C-228 Brown Hall contains 28 pentium multimedia computers, a multimedia (data/video) projection system and mounted TV/VCR, and a dry-erase board. </p>

<p>The Electronic Classroom in C-223 Brown Hall provides instructional/open access support for liberal arts and computer science courses. It also houses hardware resources for NSU's Cisco Certified Networking Academy. The facility contains 30 pentium multimedia computers, a multimedia (data/video) projection system and mounted TV/VCR, and a dry-erase board.</p>

<p>Stations in both classrooms have access to the Internet and a variety of software.</p>

</ElectrncClassroom>

<QuickLinks>

<Part1>

<Title> Computer Science </Title>

<OptionsTitle>Program Options</OptionsTitle>

<p1>-Computer Science(General Program)</p1>

<p2>-Computer Engineering</p2>

<p3>- Information Systems</p3>

<p4>-Computer Science (DNIMAS- General Option)</p4>

<p5>-Computer Science (DNIMAS-Engineering

Option)</p5>

<p6>Courses</p6>

</Part1>

<Part2>

<ST>Science and Technology</ST>

<Accreditations>Accreditations</Accreditations>

<Degrees>Degrees</Degrees>

<DNIMAS>DNIMAS</DNIMAS>

<Deprtmnts>

<DeprtmntTitle>Departments</DeprtmntTitle>

<AlliedHealth>Allied

Health</AlliedHealth>

<Biology>Biology</Biology>

<Chemistry>Chemistry</Chemistry>

<ComputerScience>ComputerScience</ComputerScience>

<Mathematics>Mathematics</Mathematics>

<Nursing>Nursing</Nursing>

<Physics-Engineering>Physics and Engineering</Physics-Engineering>

<Technology>Technology</Technology>

</Deprtmnts>

<Schools>Schools of NSU</Schools>

```

</Part2>
</QuickLinks>
<Curriculums>
  <csGeneral1>
    <a>FRS 100 Freshman Seminar 0 cr hr</a>
    <b>CHM 221/L or PHY 152/L 4 cr hr </b>
    <c>CHM 222/L or PHY 153/L 4 cr hrs </c>
    <d>CSC 169 Introduction to Computer Science 3 cr hr </d>
    <e>CSC 170 Computer Programming I 3 cr hrs </e>
    <f>CSC 192 Introduction to the Internet 1 cr hr </f>
    <g>ENG 101 Communication Skills I 3 cr hrs </g>
    <h>ENG 102 Communication Skills II 3 cr hrs </h>
    <i>HED 100 Personal and Community Health 2 cr hrs </i>
    <j>MTH 153 College Algebra and Trigonometry 3 cr hrs </j>
    <k>MTH 184 Calculus I 4 cr hrs </k>
    <l>PED 100 Fundamentals of Fitness for Life 1 cr hrs </l>
    <m>TOTAL 31 cr hrs </m>
  </csGeneral1>
  <csGeneral2>
    <p1>BIO 110, PHY 152, or CHM 221 and their labs 4 cr hrs
  </p1>
    <p2>CSC 260 Computer Programming II 3 cr hrs </p2>
    <p3>CSC 268 Computer Organization and Assembly
    Language Programming 3 cr hrs </p3>
    <p4>CSC 270 Discrete Structures 3 cr hrs </p4>
    <p5>ENG 303 Technical Writing 3 cr hrs </p5>
    <p6>HIS 100 or 101 History 3 cr hrs </p6>
    <p7>MTH 251 Calculus II 4 cr hrs </p7>
    <p8>MTH 300 Linear Algebra 3 cr hrs </p8>
    <p9>BIO 110 or PHY 152 or CHM 221 and Lab 4 cr hr
  </p9>
    <p10>SCM 285 Principles of Speech 3 cr hrs </p10>
    <p11>General Elective (200 level or above) 3 cr hr </p11>
    <p12>TOTAL 29 cr hrs </p12>
  </csGeneral2>
  <csGeneral3>
    <p1>Computer Science Elective(300 level or above) 6 cr hrs
  </p1>
    <p2>Humanities or Foreign Language 6 cr hr </p2>
    <p3>CSC 295 Java Applications Programming 3 cr hrs </p3>
    <p4>CED 350 Cooperative Education or Restricted Elective 3
    cr hrs </p4>
    <p5>CSC 361 Survey of Programming Languages 3 cr hrs
  </p5>
    <p6>CSC 372 Data Structures 3 cr hrs </p6>
    <p7>ECN 211 or 212 Principles of Economics 3 cr hrs </p7>
    <p8>MTH 351 Probability and Statistics I 3 cr hrs </p8>
    <p9>TOTAL 30 cr hrs </p9>
  </csGeneral3>
  <csGeneral4>
    <p1>Afro-American Perspective Elective 3 cr hrs </p1>
    <p2>Business or Economics Elective (300 level or above,
    excluding MSY 3XX or MSY 4XX) 3 cr hrs </p2>
    <p3>Computer Science Elective (300 level or above) 3 cr hrs
  </p3>
    <p4>Restricted Computer Science Electives 6 cr hrs </p4>

```

<p5>Restricted Mathematics Elective 3 cr hr </p5>
 <p6>CSC 430 Data Communications 3 cr hr </p6>
 <p7>CSC 464 Operating Systems 3 cr hrs </p7>
 <p8>CSC 468 Computer Architecture 3 cr hrs </p8>
 <p9>CSC 498 Computer Science Seminar I 1 cr hr </p9>
 <p10>CSC 499 Computer Science Seminar II 2 cr hrs </p10>
 <p11>TOTAL 30 cr hrs </p11>
 </csGeneral4>
 <cmptr_engnr1>
 <p1>FRS 100 Freshman Seminar 0 cr hr </p1>
 <p2>CSC 169 Introduction to Computer Science 3 cr hr </p2>
 <p3>CSC 170 Computer Programming I 3 cr hrs </p3>
 <p4>CSC 192 Introduction to the Internet 1 cr hr </p4>
 <p5>ENG 101 Communication Skills I 3 cr hrs </p5>
 <p6>ENG 102 Communication Skills II 3 cr hr </p6>
 <p7>HED 100 Personal and Community Health 2 cr hrs </p7>
 <p8>MTH 153 College Algebra and Trigonometry 3 cr hrs
 </p8>
 <p9>MTH 184 Calculus I 4 cr hrs </p9>
 <p10>PED 100 Fundamentals of Fitness for Life 1 cr hrs
 </p10>
 <p11>PHY 152 General Physics I 3 cr hrs </p11>
 <p12>PHY 152L General Physics I Lab 1 cr hrs </p12>
 <p13>PHY 153 General Physics II 3 cr hrs </p13>
 <p14>PHY 153L General Physics II Lab 1 cr hr </p14>
 <p15>TOTAL 31 cr hrs </p15>
 </cmptr_engnr1>
 <cmptr_engnr2>
 <p1>CSC 260 Computer Programming II 3 cr hrs </p1>
 <p2>CSC 268 Computer Organization and Assembly
 Language Programming 3 cr hrs </p2>
 <p3>CSC 270 Discrete Structures 3 cr hrs </p3>
 <p4>EEN 230 Electrical Network Theory 3 cr hrs </p4>
 <p5>EEN 231 Electrical Network Theory Lab 1 cr hrs </p5>
 <p6>CSC 270 Discrete Structures 3 cr hrs </p6>
 <p7>HIS 100/or 101 History 3 cr hrs </p7>
 <p8>MTH 251 Calculus II 4 cr hrs </p8>
 <p9>MTH 252 Calculus III 4 cr hr </p9>
 <p10>SCM 285 Principles of Speech 3 cr hrs </p10>
 <p11>TOTAL 30 cr hrs </p11>
 </cmptr_engnr2>
 <cmptr_engnr3>
 <p1>Group 1 Elective 3 cr hrs </p1>
 <p2>Humanities or Foreign Language 6 cr hr </p2>
 <p3>BIO 110,PHY 152, or CHM 221 and their Labs 4 cr hr
 </p3>
 <p4>MTH 372 Differential Equations 3 cr hrs </p4>
 <p5>CSC 295 Java Applications Programming 3 cr hrs </p5>
 <p6>CSC 361 Survey of Programming Language 3 cr hrs
 </p6>
 <p7>CSC 372 Data Structure 3 cr hrs </p7>
 <p8>EEN 309/311 Electronics I and Lab 4 cr hrs </p8>
 <p9>MTH 351 Probability and Statistics I 3 cr hrs </p9>
 <p10>TOTAL 32 cr hrs </p10>
 </cmptr_engnr3>
 <cmptr_engnr4>

<p1>Business or Economics Elective (200 level or above,
 excluding ASM or MSY courses 3 cr hrs</p1>
 <p3>Computer Engineering Emphasis Elective 3 cr hrs </p3>
 <p4>Computer Science Group 2 Electives) 3 cr hrs </p4>
 <p5>Afro-American Perspective Elective 3 cr hrs </p5>
 <p6>CSC 430 Data Communications 3 cr hr </p6>
 <p7>CSC 464 Operating Systems 3 cr hr </p7>
 <p8>CSC 498 Computer Science Seminar I 1 cr hrs </p8>
 <p9>CSC 499 Computer Science Seminar II 2 cr hrs </p9>
 <p10>CSC 468 Computer Architecture 3 cr hr </p10>
 <p11>EEN 444/445 Digital Electronics Logic Design and Lab
 4 cr hrs </p11>
 <p12>TOTAL 28 cr hrs </p12>
</cmptr_engnr4>
<info_systems1>
 <p1>FRS 100 Freshman Seminar 0 cr hr </p1>
 <p2>Laboratory Science Electives 8 cr hr </p2>
 <p3>CSC 169 Introduction to Computer Science 3 cr hrs
</p3>
 <p4>CSC 170 Computer Programming I 3 cr hr </p4>
 <p5>CSC 192 Introduction to the Internet 1 cr hrs </p5>
 <p6>ENG 101 Communication Skills I 3 cr hr </p6>
 <p7>ENG 102 Communication Skills II 3 cr hrs </p7>
 <p8>HED 100 Health Education 2 cr hrs </p8>
 <p9>MTH 153 College Algebra and Trigonometry 3 cr hrs
</p9>
 <p10>MTH 184 Calculus I 4 cr hrs </p10>
 <p11>PED 100 Fundamentals of Fitness for Life 1 cr hrs
</p11>
 <p12>TOTAL< 31 cr hrs </p12>
</info_systems1>
<info_systems2>
 <p1>CSC 260 Computer Programming II 3 cr hrs </p1>
 <p2>CSC 268 Computer Organization and Assembly
 Language Programming 3 cr hrs </p2>
 <p3>CSC 270 Discrete Structures 3 cr hrs </p3>
 <p4>ECN 211/or 212 Principles of Economics 3 cr hrs </p4>
 <p5>ECN 220 Economics and Business Statistics 3 cr hrs
</p5>
 <p6>HIS 100/or 101 History 3 cr hrs </p6>
 <p7>MTH 251 Calculus II 4 cr hrs </p7>
 <p8>MTH 300 Linear Algebra 3 cr hrs </p8>
 <p9>SCM 285 Principles of Speech 3 cr hr </p9>
 <p10>TOTAL 28 cr hrs </p10>
</info_systems2>
<info_systems3>
 <p1>Business Elective (see list below) 6 cr hrs </p1>
 <p2>Humanities or Foreign Language 6 cr hr </p2>
 <p3>Computer Science Group 1 Elective 3 cr hr </p3>
 <p4>CSC 295 Java Applications Programming 3 cr hrs </p4>
 <p5>CSC 320 CIS: Analysis, Design and Implementation 3 cr
 hrs </p5>
 <p6>CSC 360 Interface Design and Implementation 3 cr hrs
</p6>

<p7>CSC 361 Survey of Programming Languages 3 cr hrs
</p7>

<p8>CSC 372 Data Structures 3 cr hrs </p8>

<p9>CSC 380 Software Engineering 3 cr hrs </p9>

<p10>PSY 210 Introduction to Psychology 3 cr hr </p10>

<p11>TOTAL 36 cr hrs </p11>

</info_systems3>

<info_systems4>

<p1>African-American Perspective Elective 3 cr hrs </p1>

<p2>Computer Science Group 2 Elective 3 cr hrs </p2>

<p3>General Elective 1 cr hrs </p3>

<p4>ENG 303 Technical Writing 3 cr hrs </p4>

<p5>CSC 420 Database Principles and Design 3 cr hr </p5>

<p6>CSC 422 Database Implementation 3 cr hr </p6>

<p7>CSC 430 Data Communications 3 cr hrs </p7>

<p8>CSC 464 Operating Systems 3 cr hrs </p8>

<p9>CSC 468 Computer Architecture 3 cr hr </p9>

<p10>CSC 498 Computer Science Seminar 1 cr hrs </p10>

<p11>CSC 499 Computer Science Seminar 2 cr hrs </p11>

<p12>TOTAL 28 cr hrs </p12>

</info_systems4>

<CS_DNIMAS1>

<p1>APS 110 Applied Sciences seminar 1 cr hr </p1>

<p2>APS 111 Applied Sciences seminar 1 cr hr </p2>

<p3>CHM 223H General Chemistry I 4 cr hrs </p3>

<p4>CHM 221L General Chemistry I Lab 1 cr hr </p4>

<p5>CHM 224H General Chemistry II 4 cr hrs </p5>

<p6>CHM 222L General Chemistry II Lab 1 cr hr </p6>

<p7>CSC 169H Introduction to Computer Science 3 cr hrs

</p7>

<p8>CSC 170H Computer Programming I 3 cr hrs </p8>

<p9>ENG 101H Communication Skills I 3 cr hrs </p9>

<p10>ENG 102H Communication Skills II 3 cr hrs </p10>

<p11>HED 100 Personal and Community Health 2 cr hrs

</p11>

<p12>MTH 184H Calculus I 4 cr hrs </p12>

<p13>MTH 251H Calculus II 4 cr hrs </p13>

<p14>PED 100 Fundamentals of Fitness for Life 1 cr hr

</p14>

<p15>TOTAL 35 cr hrs </p15>

</CS_DNIMAS1>

<CS_DNIMAS2>

<p1>Humanities Elective 3 cr hrs </p1>

<p2>APS 210 Applied Sciences Seminar 1 cr hr </p2>

<p3>APS 211 Applied Sciences Seminar 1 cr hr </p3>

<p4>CSC 260 Computer Programming II 3 cr hrs </p4>

<p5>CSC 268 Computer Organization and Assembly
Language Programming 3 cr hrs </p5>

<p6>CSC 270 Discrete Structures 3 cr hrs </p6>

<p7>CSC 295 Java Applications Programming 3 cr hrs </p7>

<p8>MTH 252H Calculus III 4 cr hrs </p8>

<p9>PHY 250 University Physics 4 cr hrs </p9>

<p10>PHY 250L University Physics Lab 1 cr hr </p10>

<p11>PHY 251 University Physics 4 cr hrs </p11>
 <p12>PHY 251L University Physics Lab 1 cr hr </p12>
 <p13>SCM 285H Speech/Scientific Communication 3 cr hrs
 </p13>

<p14>TOTAL 34 cr hrs </p14>
 </CS_DNIMAS2>

<CS_DNIMAS3>

<p1> Computer Science Elective (300 level or above) 3 cr hrs
 </p1>

<p2>Restricted Elective (ECN 211 or ECN 212) 3 cr hrs
 </p2>

<p3>Restricted Elective (Business or Economics) (300 or
 level or above) 3 cr hrs </p3>
 <p4>Social Science Elective 3 cr hrs </p4>
 <p5>APS 310 Applied Sciences Seminar 1 cr hr </p5>
 <p6>APS 311 Applied Sciences Seminar 1 cr hr </p6>
 <p7>APS 350 Scientific Instrumentation 3 cr hrs </p7>
 <p8>CSC 372 Data Structures 3 cr hrs </p8>
 <p9>CSC 361 Survey of Programming Language 3 cr hrs
 </p9>

<p10>CSC 468 Computer Architecture 3 cr hrs </p10>
 <p11>MTH 300 Linear Algebra 3 cr hrs </p11>
 <p12>MTH 351 Probability and Statistics I 3 cr hrs </p12>
 <p13>TOTAL 32 cr hrs </p13>
 </CS_DNIMAS3>

<CS_DNIMAS4>

<p1> Foreign Language Elective 3 cr hrs </p1>
 <p2>Humanities Electives 3 cr hrs </p2>
 <p3>Computer Science Elective (300 level or above) 3 cr hrs
 </p3>

<p4>APS 410 Applied Sciences seminar 1 cr hr </p4>
 <p5>CSC 430 Data Communication 3 cr hrs </p5>
 <p6>CSC 464 Operating Systems 3 cr hrs </p6>
 <p7>CSC 470 Artificial Intelligence or CSC 369 or 496 3 cr
 hrs </p7>

<p8>CSC 380 Software Engineering or CSC 480 3 cr hrs
 </p8>

<p9>CSC 498 Computer Science Seminar 1 cr hr </p9>
 <p10>CSC 499 Computer Science Seminar 2 cr hrs </p10>
 <p11>MTH 384Math Modeling or MTH 352 or MTH 481 3
 cr hrs </p11>
 <p12>TOTAL 28 cr hrs </p12>
 </CS_DNIMAS4>

<CS_DNIMAS_Engnr1>

<p1>APS 110 Applied Science Seminar 1 cr hr</p1>
 <p2>APS 111 Applied Science Seminar 1 cr hr</p2>
 <p3>CHM 223H/224H General Chemistry I, II 8 cr hrs </p3>
 <p4>CHM 221L/222L General Chemistry Lab2 cr hrs</p4>
 <p5>CSC 169 Introduction to Computer Science 3 cr hrs
 </p5>

<p6>CSC 170 Computer Programming I 3 cr hrs </p6>
 <p7>CSC 192 Introduction to the Internet 1 cr hr</p7>

hrs</p8>

<p8>ENG 101H/102H Communication Skills I, II 6 cr

<p9>MTH 184H Calculus I 4 cr hrs </p9>

<p10>MTH 251H Calculus II 4 cr hrs </p10>

<p11>HED 100 Personal and Community Health 2 cr hrs

</p11>

<p12>PED 100 Fund. of Fitness for Life 1 cr hr </p12>

<p13>TOTAL 36 cr hrs</p13>

</CS_DNIMAS_Engnr1>

<CS_DNIMAS_Engnr2>

<p1>APS 210 Applied Science Seminar 1 cr hr </p1>

<p2>APS 211 Applied Science Seminar 1 cr hr </p2>

<p3>CSC 260 Computer Programming II 3 cr hrs </p3>

<p4>CSC 270 Discrete Structures 3 cr hrs </p4>

<p5>CSC 268 Computer Organization and Assembly Language 3 cr hrs </p5>

<p6>MTH 252H Calculus III 4 cr hrs </p6>

<p7>MTH 300 Linear Algebra 3 cr hrs </p7>

<p8>PHY 250H/251H University Physics 8 cr hrs </p8>

<p9>PHY 250L/251L University Physics Laboratory 2 cr hrs

</p9>

<p10>SCM 285H Speech/Scientific Communication 3 cr hrs

</p10>

<p11>Humanities Elective 3 cr hrs</p11>

<p12>TOTAL 34 cr hrs </p12>

</CS_DNIMAS_Engnr2>

<CS_DNIMAS_Engnr3>

<p1>APS 310 Applied Science Seminar 1 cr hr </p1>

<p2>APS 311 Applied Science Seminar 1 cr hr </p2>

<p3>CSC 295 Java Applications Programming 3 cr hrs </p3>

<p4>CSC 361 Survey of Programming Language 3 cr

hrs</p4>

<p5>CSC 380 Software Engineering or CSC 360 or CSC 480 3 cr hrs </p5>

<p6>CSC 372 Data Structures 3 cr hrs </p6>

<p7>MTH 351 Probability & Statistics 3 cr hrs</p7>

<p8>ECN 211 Principles of Economics 3 cr hrs </p8>

<p9>EEN 230 Electrical Network Theory 3 cr hrs </p9>

<p10>EEN 231 Electrical Network Theory Lab 1 cr hr</p10>

<p11>EEN 309 Engineering Electronics I 3 cr hrs </p11>

<p12>EEN 311 Engineering Electronics I Lab 1 cr hr </p12>

<p13>Cultural Elective 3 cr hrs </p13>

<p14>Humanities Elective 3 cr hrs </p14>

<p15>TOTAL 34 cr hrs</p15>

</CS_DNIMAS_Engnr3>

<CS_DNIMAS_Engnr4>

<p1>APS 410 Applied Science Seminar 1 cr hr </p1>

<p2>EEN 455 Control Systems Analysis, or EEN 452 Communications Systems II, or EEN 480 6 cr hrs</p2>

<p3>CSC 468 Computer Architecture 3 cr hrs </p3>

<p4>CSC 470 Artificial Intelligence or CSC 369 Theory of Computation or CSC 496 Compile Construction 3 cr hrs</p4>

<p5>CSC 430 Data Communications 3 cr hrs </p5>

```

        <p6>CSC 498 Computer Science Seminar I 1 cr hr</p6>
        <p7>CSC 499 Computer Science Seminar II 2 cr hrs</p7>
        <p8>EEN 444 Digital Electronics Logic Design 3 cr hrs </p8>
        <p9>EEN 445 Digital Electronics Logic Design 1 cr hr</p9>
        <p10>Restricted Business or Economics Elective (ACC 201
        or ECN 3XX or DSC 3XX or ENT 3XX) 3 cr hrs</p10>
        <p11>Social Science Elective 3 cr hrs </p11>
        <p12>TOTAL 29 cr hrs</p12>
    </CS_DNIMAS_Engnr4>

```

```

    </Curriculums>
</CS>

```

Appendix C

Sample stylesheet for XHTML + VoiceXML document :

---- *NSUVOICE.xsl* ----

Appendix D

Sample stylesheet for WML Document:

---- *NSU.wml* ----

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="xml" version="1.0" indent="yes"
  doctype-system="http://www.wapforum.org/DTD/wml_1.2.xml"
  doctype-public="-//WAPFORUM//DTD WML 1.2//EN" />
  <xsl:template match="/">

    <wml>
      <card title = "Computer Science Department">
        <p>
          <xsl:for-each select="CS/Head">
            <xsl:value-of select="Name"/><br/>
            <xsl:value-of select="Title"/><br/>
            <xsl:value-of select="Phone"/><br/>
          </xsl:for-each>
        </p>

        <p><b><em>emphasis:</em></b></p>
        <p><anchor><go href="#card1"/>Computer Science (General
        Program)</anchor></p>
        <p><anchor><go href="#card6"/>Computer Engineering</anchor></p>
        <p><anchor><go href="#card11"/>Information Systems</anchor></p>

```

```

    <p><anchor><go href="#card16"/>Computer Science (DNIMAS- General
    Option)</anchor></p>
    <p><anchor><go href="#card21"/>Computer Science (DNIMAS- Engineering
    Option)</anchor></p>
</card>

```

```

<card id="card1" title="CS General">
    <p><anchor><go href="#card2"/>First Year</anchor></p>
    <p><anchor><go href="#card3"/>Second Year</anchor></p>
    <p><anchor><go href="#card4"/>Third Year</anchor></p>
    <p><anchor><go href="#card5"/>Fourth Year</anchor></p>
</card>

```

```

<card id="card2" title = "CS General First Year">
    <p>
    <xsl:for-each select="CS/Curriculums/csGeneral1">
    <xsl:value-of select="a"/><br/>
    <xsl:value-of select="b"/><br/>
    <xsl:value-of select="c"/><br/>
    <xsl:value-of select="d"/><br/>
    <xsl:value-of select="e"/><br/>
    <xsl:value-of select="f"/><br/>
    <xsl:value-of select="g"/><br/>
    <xsl:value-of select="h"/><br/>
    <xsl:value-of select="i"/><br/>
    <xsl:value-of select="j"/><br/>
    <xsl:value-of select="k"/><br/>
    <xsl:value-of select="l"/><br/>
    <xsl:value-of select="m"/><br/>
    </xsl:for-each>
    </p>
</card>

```

```

<card id="card3" title = "CS General Second Year">
    <p>
    <xsl:for-each select="CS/Curriculums/csGeneral2">
    <xsl:value-of select="p1"/><br/>
    <xsl:value-of select="p2"/><br/>
    <xsl:value-of select="p3"/><br/>
    <xsl:value-of select="p4"/><br/>
    <xsl:value-of select="p5"/><br/>
    <xsl:value-of select="p6"/><br/>
    <xsl:value-of select="p7"/><br/>
    <xsl:value-of select="p8"/><br/>
    <xsl:value-of select="p9"/><br/>
    <xsl:value-of select="p10"/><br/>
    <xsl:value-of select="p11"/><br/>
    <xsl:value-of select="p12"/><br/>
    <xsl:value-of select="p13"/><br/>
    </xsl:for-each>
    </p>
</card>

```

```

<card id="card4" title = "CS General Third Year">
    <p>
    <xsl:for-each select="CS/Curriculums/csGeneral3">

```

```

        <xsl:value-of select="p1"/><br/>
        <xsl:value-of select="p2"/><br/>
        <xsl:value-of select="p3"/><br/>
        <xsl:value-of select="p4"/><br/>
        <xsl:value-of select="p5"/><br/>
        <xsl:value-of select="p6"/><br/>
        <xsl:value-of select="p7"/><br/>
        <xsl:value-of select="p8"/><br/>
        <xsl:value-of select="p9"/><br/>
    </xsl:for-each>
</p>
</card>

<card id="card5" title = "CS General Fourth Year">
    <p>
        <xsl:for-each select="CS/Curriculums/csGeneral4">
            <xsl:value-of select="p1"/><br/>
            <xsl:value-of select="p2"/><br/>
            <xsl:value-of select="p3"/><br/>
            <xsl:value-of select="p4"/><br/>
            <xsl:value-of select="p5"/><br/>
            <xsl:value-of select="p6"/><br/>
            <xsl:value-of select="p7"/><br/>
            <xsl:value-of select="p8"/><br/>
            <xsl:value-of select="p9"/><br/>
            <xsl:value-of select="p10"/><br/>
            <xsl:value-of select="p11"/><br/>
        </xsl:for-each>
    </p>
</card>

<card id="card6" title="Computer Engineering">
    <p><a href="#card7">First Year</a></p>
    <p><a href="#card8">Second Year</a></p>
    <p><a href="#card9">Third Year</a></p>
    <p><a href="#card10">Fourth Year</a></p>
</card>

<card id="card7" title = "Computer Engineering First Year">
    <p>
        <xsl:for-each select="CS/Curriculums/cmptr_engnr1">
            <xsl:value-of select="p1"/><br/>
            <xsl:value-of select="p2"/><br/>
            <xsl:value-of select="p3"/><br/>
            <xsl:value-of select="p4"/><br/>
            <xsl:value-of select="p5"/><br/>
            <xsl:value-of select="p6"/><br/>
            <xsl:value-of select="p7"/><br/>
            <xsl:value-of select="p8"/><br/>
            <xsl:value-of select="p9"/><br/>
            <xsl:value-of select="p10"/><br/>
            <xsl:value-of select="p11"/><br/>
            <xsl:value-of select="p12"/><br/>
            <xsl:value-of select="p13"/><br/>
            <xsl:value-of select="p14"/><br/>
            <xsl:value-of select="p15"/><br/>
        </xsl:for-each>
    </p>

```

```

        </xsl:for-each>
    </p>
</card>

<card id="card8" title = "Computer Engineering Second Year">
    <p>
        <xsl:for-each select="CS/Curriculums/cmptr_engnr2">
            <xsl:value-of select="p1"/><br/>
            <xsl:value-of select="p2"/><br/>
            <xsl:value-of select="p3"/><br/>
            <xsl:value-of select="p4"/><br/>
            <xsl:value-of select="p5"/><br/>
            <xsl:value-of select="p6"/><br/>
            <xsl:value-of select="p7"/><br/>
            <xsl:value-of select="p8"/><br/>
            <xsl:value-of select="p9"/><br/>
            <xsl:value-of select="p10"/><br/>
            <xsl:value-of select="p11"/><br/>
        </xsl:for-each>
    </p>
</card>

<card id="card9" title = "Computer Engineering Third Year">
    <p>
        <xsl:for-each select="CS/Curriculums/cmptr_engnr3">
            <xsl:value-of select="p1"/><br/>
            <xsl:value-of select="p2"/><br/>
            <xsl:value-of select="p3"/><br/>
            <xsl:value-of select="p4"/><br/>
            <xsl:value-of select="p5"/><br/>
            <xsl:value-of select="p6"/><br/>
            <xsl:value-of select="p7"/><br/>
            <xsl:value-of select="p8"/><br/>
            <xsl:value-of select="p9"/><br/>
            <xsl:value-of select="p10"/><br/>
        </xsl:for-each>
    </p>
</card>

<card id="card10" title = "Computer Engineering Fourth Year">
    <p>
        <xsl:for-each select="CS/Curriculums/cmptr_engnr4">
            <xsl:value-of select="p1"/><br/>
            <xsl:value-of select="p2"/><br/>
            <xsl:value-of select="p3"/><br/>
            <xsl:value-of select="p4"/><br/>
            <xsl:value-of select="p5"/><br/>
            <xsl:value-of select="p6"/><br/>
            <xsl:value-of select="p7"/><br/>
            <xsl:value-of select="p8"/><br/>
            <xsl:value-of select="p9"/><br/>
            <xsl:value-of select="p10"/><br/>
            <xsl:value-of select="p11"/><br/>
            <xsl:value-of select="p12"/><br/>
        </xsl:for-each>
    </p>

```

```

</card>

<card id="card11" title="Information Systems">
  <p><a href="#card12">First Year</a></p>
  <p><a href="#card13">Second Year</a></p>
  <p><a href="#card14">Third Year</a></p>
  <p><a href="#card15">Fourth Year</a></p>
</card>

<card id="card12" title = "Information Systems First Year">
  <p>
    <xsl:for-each select="CS/Curriculums/info_systems1">
      <xsl:value-of select="p1"/><br/>
      <xsl:value-of select="p2"/><br/>
      <xsl:value-of select="p3"/><br/>
      <xsl:value-of select="p4"/><br/>
      <xsl:value-of select="p5"/><br/>
      <xsl:value-of select="p6"/><br/>
      <xsl:value-of select="p7"/><br/>
      <xsl:value-of select="p8"/><br/>
      <xsl:value-of select="p9"/><br/>
      <xsl:value-of select="p10"/><br/>
      <xsl:value-of select="p11"/><br/>
      <xsl:value-of select="p12"/><br/>
    </xsl:for-each>
  </p>
</card>

<card id="card13" title = "Information Systems Second Year">
  <p>
    <xsl:for-each select="CS/Curriculums/info_systems2">
      <xsl:value-of select="p1"/><br/>
      <xsl:value-of select="p2"/><br/>
      <xsl:value-of select="p3"/><br/>
      <xsl:value-of select="p4"/><br/>
      <xsl:value-of select="p5"/><br/>
      <xsl:value-of select="p6"/><br/>
      <xsl:value-of select="p7"/><br/>
      <xsl:value-of select="p8"/><br/>
      <xsl:value-of select="p9"/><br/>
      <xsl:value-of select="p10"/><br/>
    </xsl:for-each>
  </p>
</card>

<card id="card14" title = "Information Systems Third Year">
  <p>
    <xsl:for-each select="CS/Curriculums/info_systems3">
      <xsl:value-of select="p1"/><br/>
      <xsl:value-of select="p2"/><br/>
      <xsl:value-of select="p3"/><br/>
      <xsl:value-of select="p4"/><br/>
      <xsl:value-of select="p5"/><br/>
      <xsl:value-of select="p6"/><br/>
      <xsl:value-of select="p7"/><br/>
      <xsl:value-of select="p8"/><br/>
    </xsl:for-each>
  </p>
</card>

```

```

        <xsl:value-of select="p9"/><br/>
        <xsl:value-of select="p10"/><br/>
        <xsl:value-of select="p11"/><br/>
    </xsl:for-each>
</p>
</card>

<card id="card15" title = "Information Systems Fourth Year">
    <p>
        <xsl:for-each select="CS/Curriculums/info_systems4">
            <xsl:value-of select="p1"/><br/>
            <xsl:value-of select="p2"/><br/>
            <xsl:value-of select="p3"/><br/>
            <xsl:value-of select="p4"/><br/>
            <xsl:value-of select="p5"/><br/>
            <xsl:value-of select="p6"/><br/>
            <xsl:value-of select="p7"/><br/>
            <xsl:value-of select="p8"/><br/>
            <xsl:value-of select="p9"/><br/>
            <xsl:value-of select="p10"/><br/>
            <xsl:value-of select="p11"/><br/>
            <xsl:value-of select="p12"/><br/>
        </xsl:for-each>
    </p>
</card>

<card id="card16" title="CS DNIMAS">
    <p><a href="#card17">First Year</a></p>
    <p><a href="#card18">Second Year</a></p>
    <p><a href="#card19">Third Year</a></p>
    <p><a href="#card20">Fourth Year</a></p>
</card>

<card id="card17" title = "CS DNIMAS First Year">
    <p>
        <xsl:for-each select="CS/Curriculums/CS_DNIMAS1">
            <xsl:value-of select="p1"/><br/>
            <xsl:value-of select="p2"/><br/>
            <xsl:value-of select="p3"/><br/>
            <xsl:value-of select="p4"/><br/>
            <xsl:value-of select="p5"/><br/>
            <xsl:value-of select="p6"/><br/>
            <xsl:value-of select="p7"/><br/>
            <xsl:value-of select="p8"/><br/>
            <xsl:value-of select="p9"/><br/>
            <xsl:value-of select="p10"/><br/>
            <xsl:value-of select="p11"/><br/>
            <xsl:value-of select="p12"/><br/>
            <xsl:value-of select="p13"/><br/>
            <xsl:value-of select="p14"/><br/>
            <xsl:value-of select="p15"/><br/>
        </xsl:for-each>
    </p>
</card>

<card id="card18" title = "CS DNIMAS Second Year">

```

```

<p>
  <xsl:for-each select="CS/Curriculums/CS_DNIMAS2">
    <xsl:value-of select="p1"/><br/>
    <xsl:value-of select="p2"/><br/>
    <xsl:value-of select="p3"/><br/>
    <xsl:value-of select="p4"/><br/>
    <xsl:value-of select="p5"/><br/>
    <xsl:value-of select="p6"/><br/>
    <xsl:value-of select="p7"/><br/>
    <xsl:value-of select="p8"/><br/>
    <xsl:value-of select="p9"/><br/>
    <xsl:value-of select="p10"/><br/>
    <xsl:value-of select="p11"/><br/>
    <xsl:value-of select="p12"/><br/>
    <xsl:value-of select="p13"/><br/>
  </xsl:for-each>
</p>
</card>

```

```

<card id="card19" title = "CS DNIMAS Third Year">
  <p>
    <xsl:for-each select="CS/Curriculums/CS_DNIMAS3">
      <xsl:value-of select="p1"/><br/>
      <xsl:value-of select="p2"/><br/>
      <xsl:value-of select="p3"/><br/>
      <xsl:value-of select="p4"/><br/>
      <xsl:value-of select="p5"/><br/>
      <xsl:value-of select="p6"/><br/>
      <xsl:value-of select="p7"/><br/>
      <xsl:value-of select="p8"/><br/>
      <xsl:value-of select="p9"/><br/>
      <xsl:value-of select="p10"/><br/>
      <xsl:value-of select="p11"/><br/>
      <xsl:value-of select="p12"/><br/>
      <xsl:value-of select="p13"/><br/>
    </xsl:for-each>
  </p>
</card>

```

```

<card id="card20" title = "CS DNIMAS Fourth Year">
  <p>
    <xsl:for-each select="CS/Curriculums/CS_DNIMAS4">
      <xsl:value-of select="p1"/><br/>
      <xsl:value-of select="p2"/><br/>
      <xsl:value-of select="p3"/><br/>
      <xsl:value-of select="p4"/><br/>
      <xsl:value-of select="p5"/><br/>
      <xsl:value-of select="p6"/><br/>
      <xsl:value-of select="p7"/><br/>
      <xsl:value-of select="p8"/><br/>
      <xsl:value-of select="p9"/><br/>
      <xsl:value-of select="p10"/><br/>
      <xsl:value-of select="p11"/><br/>
      <xsl:value-of select="p12"/><br/>
    </xsl:for-each>
  </p>

```

</card>

```
<card id="card21" title="CS DNIMAS (Engineering)">
  <p><a href="#card22">First Year</a></p>
  <p><a href="#card23">Second Year</a></p>
  <p><a href="#card24">Third Year</a></p>
  <p><a href="#card25">Fourth Year</a></p>
</card>
```

```
<card id="card22" title = "CS DNIMAS (Engineering) First Year">
  <p>
    <xsl:for-each select="CS/Curriculums/CS_DNIMAS_Engnr1">
      <xsl:value-of select="p1"/><br/>
      <xsl:value-of select="p2"/><br/>
      <xsl:value-of select="p3"/><br/>
      <xsl:value-of select="p4"/><br/>
      <xsl:value-of select="p5"/><br/>
      <xsl:value-of select="p6"/><br/>
      <xsl:value-of select="p7"/><br/>
      <xsl:value-of select="p8"/><br/>
      <xsl:value-of select="p9"/><br/>
      <xsl:value-of select="p10"/><br/>
      <xsl:value-of select="p11"/><br/>
      <xsl:value-of select="p12"/><br/>
      <xsl:value-of select="p13"/><br/>
    </xsl:for-each>
  </p>
</card>
```

```
<card id="card23" title = "CS DNIMAS (Engineering) Second Year">
  <p>
    <xsl:for-each select="CS/Curriculums/CS_DNIMAS_Engnr2">
      <xsl:value-of select="p1"/><br/>
      <xsl:value-of select="p2"/><br/>
      <xsl:value-of select="p3"/><br/>
      <xsl:value-of select="p4"/><br/>
      <xsl:value-of select="p5"/><br/>
      <xsl:value-of select="p6"/><br/>
      <xsl:value-of select="p7"/><br/>
      <xsl:value-of select="p8"/><br/>
      <xsl:value-of select="p9"/><br/>
      <xsl:value-of select="p10"/><br/>
      <xsl:value-of select="p11"/><br/>
      <xsl:value-of select="p12"/><br/>
    </xsl:for-each>
  </p>
</card>
```

```
<card id="card24" title = "CS DNIMAS (Engineering) Third Year">
  <p>
    <xsl:for-each select="CS/Curriculums/CS_DNIMAS_Engnr3">
      <xsl:value-of select="p1"/><br/>
      <xsl:value-of select="p2"/><br/>
      <xsl:value-of select="p3"/><br/>
      <xsl:value-of select="p4"/><br/>
      <xsl:value-of select="p5"/><br/>
    </xsl:for-each>
  </p>
</card>
```

```

        <xsl:value-of select="p6"/><br/>
        <xsl:value-of select="p7"/><br/>
        <xsl:value-of select="p8"/><br/>
        <xsl:value-of select="p9"/><br/>
        <xsl:value-of select="p10"/><br/>
        <xsl:value-of select="p11"/><br/>
        <xsl:value-of select="p12"/><br/>
        <xsl:value-of select="p13"/><br/>
        <xsl:value-of select="p14"/><br/>
        <xsl:value-of select="p15"/><br/>
    </xsl:for-each>
</p>
</card>

<card id="card25" title = "CS DNIMAS (Engineering) Fourth Year">
    <p>
        <xsl:for-each select="CS/Curriculums/CS_DNIMAS_Engnr4">
            <xsl:value-of select="p1"/><br/>
            <xsl:value-of select="p2"/><br/>
            <xsl:value-of select="p3"/><br/>
            <xsl:value-of select="p4"/><br/>
            <xsl:value-of select="p5"/><br/>
            <xsl:value-of select="p6"/><br/>
            <xsl:value-of select="p7"/><br/>
            <xsl:value-of select="p8"/><br/>
            <xsl:value-of select="p9"/><br/>
            <xsl:value-of select="p10"/><br/>
            <xsl:value-of select="p11"/><br/>
            <xsl:value-of select="p12"/><br/>
        </xsl:for-each>
    </p>
</card>
</wml>
</xsl:template>
</xsl:stylesheet>

```

Appendix E

Apache Tomcat and Cocoon Installation

- **Download and Install**

- ✓ J2SDK1.4.1_01 or latest (<http://java.sun.com/j2se/1.4/download.html>)
- ✓ Apache Tomcat-4.1.12 or latest (<http://jakarta.apache.org/tomcat/>)

- ✓ Apache Cocoon 2.0.3 or latest binary distribution (<http://xml.apache.org/cocoon/dist/>)

- **Set Environment Variables**

Goto: Control Panel -> System -> Advanced -> Environment Variables

Add Variables:

JAVA_HOME = \ path \ to \ java

Example: JAVA_HOME = C:\j2sdk1.4.01

TOMCAT_HOME = \ path \ to \ tomcat

Example: TOMCAT_HOME = C:\jakarta-tomcat-4.1.12

PATH = \ path \ to \ java \ bin

Example: PATH = C:\j2sdk1.4.01\bin

- Copy the cocoon.war file from C:\cocoon-2.0.3 directory into TOMCAT_HOME\webapps directory.
- Start tomcat by going to the TOMCAT_HOME\bin directory and running the startup.bat script.
- Open your internet browser and enter: <http://localhost:8080/cocoon/>
- If you correctly followed the above procedure, you should see cocoon's welcome page
- Congratulations! Cocoon is now up and running.

To stop tomcat, simply go back to the TOMCAT_HOME\bin directory, and run the shutdown.bat script or use [Ctrl]+[C].

If you have difficulties with any installation, please visit their respective sites for more information.

Appendix F

MySQL Installation

- **Download and Install**
 - ✓ MySQL-4.1.0.alpha or latest (<http://www.mysql.com/downloads/index.html>)
 - ✓ Mysql-connector-java-3.1.0-alpha or latest (<http://www.mysql.com/downloads/index.html>)
- **Set Environment Variables**

Goto: Control Panel -> System -> Advanced -> Environment Variables

Add Variables:

PATH = \ path \ to \ mysql \ bin

Example: PATH = C:\mysql\bin

CLASSPATH = \ classpath \ to \ Driver class of MySQL-Connector

Example: CLASSPATH = C:\mySQL-Connector\mysql-connector-java-3.1.0-alpha\com\mysql\jdbc

- Copy the JAR file (JAR file example: mysql-connector-java-3.1.0-alpha.jar) of your driver class into TOMCAT_HOME/Webapps/Cocoon/WEB-INF/lib and JAVA_HOME\jre\lib
- Open your command prompt and enter path to MySQL bin.
Example: C:\cd mysql\bin
- Enter mysql -u root and you should see

```
Welcome to the MySQL monitor.  Commands end with ; or
\g.
Your MySQL connection id is 23 to server version:
3.23.28-gamma-log

Type 'help;' or '\h' for help. Type '\c' to clear the
buffer

mysql>
```

If you have difficulties with your installation or connection to MySQL server, kindly visit MySQL Reference Manual (<http://www.mysql.com/doc/en/index.html>) for more information.

Appendix G

Xindice Installation

- Download the zip file from www.xindice.org or apache.org.
- Unzip the file into a suitable directory.
- Add the Xindice bin directory into your path. For example, C:\Xindicexml\xindice-1.0\bin .
- Set the JAVA_HOME environment variable to point to the location where the JDK is installed. Remember that you need Sun's Java SDK version 1.3. At this

time there appears to be problems with JDK 1.4 and JDK 1.2 or lower will not work.

- Set the XINDICE_HOME environment variable to point to the location where Xindice is installed. For example C:\Xindicexml\indice-1.0
- Add the Xindice.jar file to your CLASSPATH. For example, C:\Xindicexml\xindice-1.0\java\lib\Xindice.jar.

Note: Installation Instructions are taken from the Dev Newz site at <http://www.devnewz.com/articles/0220.html>.

Appendix H

Xindice Command Line Tools

- Creating a collection:

Creating a top level collection

```
xindice add_collection -c /db -n nsu
```

Creating a collection under an existing collection

```
xindice add_collection -c /db/nsu -n  
computer_science
```

- Deleting a collection

Deleting a top level collection (note: sub collections and documents will also be deleted)

```
xindice add_collection -c /db -n nsu
```

Deleting a sub collection

```
xindice delete_collection -c /db/nsu -n  
computer_science
```

- List Collections

Listing the Top Level Collections

```
xindice list_collections -c /db
```

Listing a Sub Collection

```
xindice list_collections -c /db/nsu
```

- Adding a Document

Adding a document to a collection with a key of "engineering"

```
xindice add_document -c /db/nsu -f  
/tmp/engineering.xml -n stones
```

**Adding a document to the root collection with an automatically
generated key**

```
xindice add_document -c /db -f /tmp/CS.xml
```

- Deleting a Document

Deleting a Document from a collection

```
xindice delete_document -c /db/nsu -n  
engineering
```

- Retrieving a Document

Retrieving a document from a collection

```
xindice retrieve_document -c /db/nsu -n engineering -f  
/tmp/engineering.xml
```

- Executing an Xpath Query against a Collection

**Run an Xpath query against all documents in the nsu collection with
"IS" node that has the type attribute = "DNIMAS"**

```
xindice xpath -c /db/nsu -q  
/IS[@type="DNIMAS"]
```

Appendix F

Java-based client/server system with a form to generate XML documents

By Cyntrica Eaton

What is it?

A client and server system written in Java communicating using a high TCP port (8000). Although this 2-tier client/server will NOT be used for implementation, it does serve the purpose of demonstrating the ability of creating a Form rendered in Java, which can generate an XML document from user input.

Client User Interface

A screenshot of a Java-based client user interface. The window has a blue title bar with a small icon on the left and standard minimize, maximize, and close buttons on the right. The main content area is a form with a light gray background. It contains two text input fields: the first is labeled "Name:" and contains the text "kung"; the second is labeled "E-mail:" and contains the text "mlkung@nsu.edu". Below the input fields is a "Submit" button.

After Data were fill in and click, the following data is received at the server:

```
<XML version10><DOCTYPE userinfo SYSTEM userinfodtd><info><name>kung</name>  
<email>mlkung@nsu.edu</email></info>
```

Client Source

```
import java.io.*;
import java.net.*;
import java.awt.event.*;
import javax.swing.*;
import java.awt.*;

public class UserDataForm extends JFrame {

    private JLabel nameLabel = new JLabel("Name: ", SwingConstants.RIGHT);
    JTextField nameField = new JTextField();

    private JLabel emailLabel = new JLabel("E-mail: ", SwingConstants.RIGHT);
    JTextField emailField = new JTextField();

    JButton submit = new JButton("Submit");

    public UserDataForm () {

        //Create the user interface

        nameField = new JTextField();
        emailField = new JTextField();

        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(3, 2));
        panel.add(nameLabel);
        panel.add(nameField);
        panel.add(emailLabel);
        panel.add(emailField);

        submit.addActionListener(new ButtonListener());
        panel.add(submit);
        setContentPane(panel);
    }

    static public void main (String args[])
    {
        UserDataForm userForm = new UserDataForm();
        userForm.pack();
        userForm.setVisible(true);
    }

    public class ButtonListener implements ActionListener {
        public void actionPerformed (ActionEvent e) {
            //
            String name = nameField.getText();
            String email = emailField.getText();

            try {
                DataOutputStream file = new DataOutputStream (new
FileOutputStream ("Source.txt"));
                file.writeBytes("<XML version10>");
                file.writeBytes("<DOCTYPE userinfo SYSTEM userinfodtd>");
            }
        }
    }
}
```

```

        file.writeBytes("<info>");
        file.writeBytes("<name>" + name + "</name>");
        file.writeBytes("\n");
        file.writeBytes("<email>" + email + "</email>");
        file.writeBytes("</info>");

        file.close();
    }
    catch (IOException ie) {
    }

    //After create file send it off
    Socket s = null;

    OutputStream out = null;
    BufferedOutputStream bufferedOut = null;
    DataOutputStream fileData = null;

    byte buf [] = new byte[2048];

    try {
        s = new Socket ("localhost", 8000);
        out = s.getOutputStream();
        bufferedOut = new BufferedOutputStream(out, 2048);
        fileData = new DataOutputStream(bufferedOut);
    }
    catch (IOException ie) {
        System.err.println("I/O Exception");
    }
    catch (Exception ee) {
        System.err.println("General Exception");
    }
    }

    int c, length;
    int ct = 0;

    try {
        File sourceFile = new File("Source.txt");

        FileInputStream fis = new FileInputStream(sourceFile);
        length = (int)sourceFile.length();

        //send length of file
        fileData.writeInt(length);
        fileData.flush();

        //send file

        while (ct < length) {
            c = fis.read();
            fileData.writeByte(c);
            ct++;
        }

        //clean up
        fileData.flush();
    }

```



```

    }
    catch (IOException e) {
        System.err.println("I/O Error ");
        System.exit(1);
    }
}

public void run () {

    InputStream readFile = null;
    BufferedInputStream bufferedReadFile = null;
    DataInputStream fileInfo = null;

    OutputStream out = null;
    DataOutputStream ServerThreadOut = null;

    try {
        readFile = clientSocket.getInputStream();
        bufferedReadFile = new BufferedInputStream(readFile, 2048);
        fileInfo = new DataInputStream(bufferedReadFile);
    }
    catch (IOException e) {e.printStackTrace();}

    byte [] c = null;

    try {

        File destinationFile = new File("Destination.txt");
        FileOutputStream fos = new FileOutputStream(destinationFile);
        int ct = 0;
        int length = 0;
        length = fileInfo.readInt();
        c = new byte[length];

        //Read in data for the entire length of the file
        while (ct < length) {
            c[ct] = fileInfo.readByte();
            ct++;
        }

        fos.write(c);
        fos.close();
    }

    catch (IOException e) {e.printStackTrace();}
    System.out.println("File Recieved");
}
}

```

Appendix G

Web Form Page to Generate XML Document from User Input By Matthew Sanders

DEPARTMENT OF
TRANSPORTATION
FEDERAL AVIATION
ADMINISTRATION

FORM APPROVED OMB No.
2120-0008
RIS: WS 8070-1
Control No.

Service Difficulty Report
AERONAUTICAL EQUIPMENT

ATA Code

MAJOR EQUIPMENT IDENTITY

Enter pertinent data	MANUFACTURER MODEL/SERIES	SERIAL NUMBER	N- Aircraft Registration Number
AIRCRAFT	<input type="text"/>	<input type="text"/>	<input type="text"/>
POWERPLANT	<input type="text"/>	<input type="text"/>	<input type="text"/>
PROPELLER	<input type="text"/>	<input type="text"/>	<input type="text"/>

PROBLEM DESCRIPTION

DATE	STATUS	CARRIER	ATA	AIRCRAFT TYPE	N-	CONTROL NO.
<input type="text"/>						

TEXT

SPECIFIC PART CAUSING PROBLEM

PART NAME	MFG PART NUMBER	PART CONDITION	PART/DEFECT
-----------	-----------------	----------------	-------------

			LOCATION
--	--	--	----------

COMPONENT? APPLIANCE ABOVE PART INSTALLED ON

Report whole hours

PART TT

PART TSO

COMP/APPL NAME

MANUFACTURER

MFG. MODEL/NUMBER

SERIAL NO.

SUBMIT BY

SUBMITTER (check one)

A B C D E F G H I

REP STA R OPE R ME CH AIR T AXI MF G FA A OTH ER Spec

P.S.L. ALERT OPER/D.O.

PREC PROC

NATURE

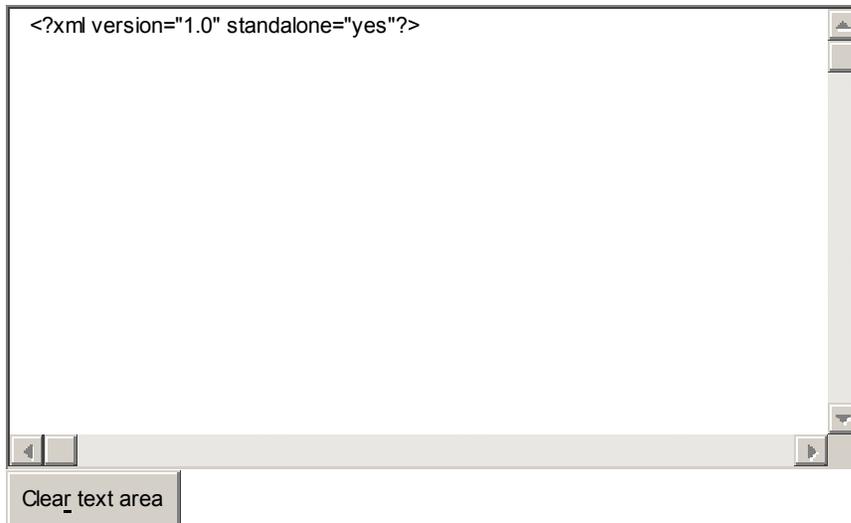
STAGE ROLL

STAT FRAME

SYS. SYS.

ADDITIONAL COMMENTS

Clear Information



Source Code

```
<html>
<head>
<title>faa form</title>
<SCRIPT>
function XMLForm(form){

    if(form.ata_f.value !="" )
    {
    form.ata_p.value = form.ata_f.value
    ata_fp = form.ata_f.value
    }
    else
    {
    ata_fp = window.prompt("The ata is a required field. Please enter the ata?","ATA")
    }

    var tag_21200008 = "<form_2120_0008 control_no=" + form.control_no_f.value + " ata="
+ ata_fp + " code=" + form.code_f.value + ">"
    var Etag_21200008 = "</form_2120_0008>"
    var tag_major_eq_id = "<major_equipment_identity>"
    var Etag_major_eq_id = "</major_equipment_identity>"
    var tag_aircraft = "<aircraft>"
    var Etag_aircraft = "</aircraft>"
    var tag_powerplant = "<powerplant>"
    var Etag_powerplant = "</powerplant>"
    var tag_propeller = "<propeller>"
    var Etag_propeller = "</propeller>"
    var tag_prob_des = "<problem_description>>"
    var Etag_prob_des = "</problem_description>"
    var tag_submitter = "<submitter>"
    var Etag_submitter = "</submitter>"
```

```

//ris: ws 8070-1 begin
form.control_no_f.value = "<control_no>" + form.control_no_f.value + "</control_no>"
form.ata_f.value = "<ata>" + ata_fp + "</ata>"
form.code_f.value = "<code>" + form.code_f.value + "</code>"
//ris: ws 8070-1 end

//major_equipment_identity begin
form.aircraft_man.value = "<aircraft_man>" + form.aircraft_man.value + "</aircraft_man>"

form.aircraft_mod.value = "<aircraft_mod>" + form.aircraft_mod.value + "</aircraft_mod>"

form.aircraft_ser.value = "<aircraft_ser>" + form.aircraft_ser.value + "</aircraft_ser>"
form.aircraft_reg_eq.value = "<aircraft_reg_eq>" + form.aircraft_reg_eq.value +
"</aircraft_reg_eq>"

form.powerplant_man.value = "<powerplant_man>" + form.powerplant_man.value +
"</powerplant_man>"
form.powerplant_mod.value = "<powerplant_mod>" + form.powerplant_mod.value +
"</powerplant_mod>"
form.powerplant_ser.value = "<powerplant_ser>" + form.powerplant_ser.value +
"</powerplant_ser>"
form.powerplant_reg_eq.value = "<powerplant_reg_eq>" +
form.powerplant_reg_eq.value + "</powerplant_reg_eq>"

form.propeller_man.value = "<propeller_man>" + form.propeller_man.value +
"</propeller_man>"
form.propeller_mod.value = "<propeller_mod>" + form.propeller_mod.value +
"</propeller_mod>"
form.propeller_ser.value = "<propeller_ser>" + form.propeller_ser.value +
"</propeller_ser>"
form.propeller_reg_eq.value = "<propeller_reg_eq>" + form.propeller_reg_eq.value +
"</propeller_reg_eq>"
//major_equipment_identity end

//problem_description begin
form.date_p.value = "<date>" + form.date_p.value + "</date>"
form.status.value = "<status>" + form.status.value + "</status>"
form.carrier.value = "<carrier>" + form.carrier.value + "</carrier>"
form.ata_p.value = "<ata>" + ata_fp + "</ata>"
form.aircraft_type.value = "<aircraft_type>" + form.aircraft_type.value + "</aircraft_type>"
form.aircraft_reg.value = "<aircraft_reg>" + form.aircraft_reg.value + "</aircraft_reg>"
form.control_no_p.value = "<control_no>" + form.control_no_p.value + "</control_no>"
form.prob_tx.value = "<probtext>" + form.prob_tx.value + "</probtext>"
form.part_name.value = "<part_name>" + form.part_name.value + "</part_name>"
form.mfg_part_num.value = "<mfg_part_num>" + form.mfg_part_num.value +
"</mfg_part_num>"
form.part_condition.value = "<part_condition>" + form.part_condition.value +
"</part_condition>"
form.part_defect_loc.value = "<part_defect_loc>" + form.part_defect_loc.value +
"</part_defect_loc>"
form.comp_app_inst.value = "<comp_appl_inst>" + form.comp_app_inst.value +
"</comp_appl_inst>"
form.part_tt.value = "<part_tt>" + form.part_tt.value + "</part_tt>"
form.part_tso.value = "<part_tso>" + form.part_tso.value + "</part_tso>"
form.comp_appl_name.value = "<comp_appl_name>" + form.comp_appl_name.value +
"</comp_appl_name>"

```

```

        form.manufacturer.value = "<manufacturer>" + form.manufacturer.value +
"</manufacturer>"
        form.mfg_model_num.value = "<mfg_model_num>" + form.mfg_model_num.value +
"</mfg_model_num>"
        form.serial_no.value = "<serial_no>" + form.serial_no.value + "</serial_no>"
//problem_description end

//submitter begin
form.submitter.value = "<submitter>" + form.submitter.value + "</submitter>"
form.carrier_ch.value = "<carrier_ch>" + form.carrier_ch.value + "</carrier_ch>"
form.rep_sta_ch.value = "<rep_sta_ch>" + form.rep_sta_ch.value + "</rep_sta_ch>"
form.oper_ch.value = "<oper_ch>" + form.oper_ch.value + "</oper_ch>"
form.mech_ch.value = "<mech_ch>" + form.mech_ch.value + "</mech_ch>"
form.air_taxi_ch.value = "<air_taxi_ch>" + form.air_taxi_ch.value + "</air_taxi_ch>"
form.mfg_ch.value = "<mfg_ch>" + form.mfg_ch.value + "</mfg_ch>"
form.faa_ch.value = "<faa_ch>" + form.faa_ch.value + "</faa_ch>"
form.spec_ch.value = "<spec_ch>" + form.spec_ch.value + "</spec_ch>"
form.psl.value = "<psl>" + form.psl.value + "</psl>"
form.alert_s.value = "<alert>" + form.alert_s.value + "</alert>"
form.oper_do.value = "<oper_do>" + form.oper_do.value + "</oper_do>"
form.prec_proc.value = "<prec_proc>" + form.prec_proc.value + "</prec_proc>"
form.nature.value = "<nature>" + form.nature.value + "</nature>"
form.stage.value = "<stage>" + form.stage.value + "</stage>"
form.stat.value = "<stat>" + form.stat.value + "</stat>"
form.roll.value = "<roll>" + form.roll.value + "</roll>"
form.frame.value = "<frame>" + form.frame.value + "</frame>"
form.sys1.value = "<sys>" + form.sys1.value + "</sys>"
form.sys2.value = "<sys>" + form.sys2.value + "</sys>"
form.comments_tx.value = "<comments_tx>" + form.comments_tx.value +
"</commets_tx>"
//submitter end

//main tag
document.XAdd.write.value += (tag_21200008 + "\n")

//major_equipment_identity      begin print out
document.XAdd.write.value += ("\t" + tag_major_eq_id + "\n")
        document.XAdd.write.value += ("\t\t" + tag_aircraft + "\n")
        document.XAdd.write.value += ("\t\t\t" + form.aircraft_man.value + "\n")
        document.XAdd.write.value += ("\t\t\t\t" + form.aircraft_mod.value + "\n")
        document.XAdd.write.value += ("\t\t\t\t" + form.aircraft_ser.value + "\n")
        document.XAdd.write.value += ("\t\t\t\t" + form.aircraft_reg_eq.value + "\n")
        document.XAdd.write.value += ("\t\t\t" + Etag_aircraft + "\n")

        document.XAdd.write.value += ("\t\t\t" + tag_powerplant + "\n")
        document.XAdd.write.value += ("\t\t\t\t" + form.powerplant_man.value + "\n")
        document.XAdd.write.value += ("\t\t\t\t\t" + form.powerplant_mod.value + "\n")
        document.XAdd.write.value += ("\t\t\t\t\t" + form.powerplant_ser.value + "\n")
        document.XAdd.write.value += ("\t\t\t\t\t" + form.powerplant_reg_eq.value + "\n")
        document.XAdd.write.value += ("\t\t\t\t" + Etag_powerplant + "\n")

        document.XAdd.write.value += ("\t\t\t\t" + tag_propeller + "\n")
        document.XAdd.write.value += ("\t\t\t\t\t" + form.propeller_man.value + "\n")
        document.XAdd.write.value += ("\t\t\t\t\t\t" + form.propeller_mod.value + "\n")
        document.XAdd.write.value += ("\t\t\t\t\t\t" + form.propeller_ser.value + "\n")
        document.XAdd.write.value += ("\t\t\t\t\t\t" + form.propeller_reg_eq.value + "\n")

```

```

        document.XAdd.write.value += ("\t\t" + Etag_propeller + "\n")
document.XAdd.write.value += ("\t" + Etag_major_eq_id + "\n")
//major_equipment_identity      end print out

//problem_description begin print out
document.XAdd.write.value += ("\t" + tag_prob_des + "\n")
    document.XAdd.write.value += ("\t\t" + form.date_p.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.status.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.carrier.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.ata_p.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.aircraft_type.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.aircraft_reg.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.control_no_p.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.prob_tx.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.part_name.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.mfg_part_num.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.part_condition.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.part_defect_loc.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.comp_app_inst.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.part_tt.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.part_tso.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.comp_appl_name.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.manufacturer.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.mfg_model_num.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.serial_no.value + "\n")
document.XAdd.write.value += ("\t" + Etag_prob_des + "\n")
//problem_description end print out

//submitter begin print out
document.XAdd.write.value += ("\t" + tag_submitter + "\n")
    document.XAdd.write.value += ("\t\t" + form.submitter.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.carrier_ch.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.rep_sta_ch.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.oper_ch.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.mech_ch.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.air_taxi_ch.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.mfg_ch.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.faa_ch.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.spec_ch.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.psl.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.alert_s.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.oper_do.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.prec_proc.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.nature.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.stage.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.stat.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.roll.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.frame.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.sys1.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.sys2.value + "\n")
    document.XAdd.write.value += ("\t\t" + form.comments_tx.value + "\n")
document.XAdd.write.value += ("\t" + Etag_submitter + "\n")
//submitter end print out

document.XAdd.write.value += (Etag_21200008 + "\n")
}

```

```

</SCRIPT>
</head>

<body bgcolor="#FFFFFF">
<form name="xform">
  <div align="center">
    <table width="95%" border="0">
      <tr>
        <td>
          <div align="center">
            <table width="75%" border="0">
              <tr>
                <td width="48%">
                  <table width="100%" border="0">
                    <tr>
                      <td>
                        <div align="center"><b>DEPARTMENT OF TRANSPORTATION</b></div>
                      </td>
                    </tr>
                    <tr>
                      <td>
                        <div align="center"><font size="2">FEDERAL AVIATION
ADMINISTRATION</font></div>
                      </td>
                    </tr>
                    <tr>
                      <td>&nbsp;</td>
                    </tr>
                    <tr>
                      <td>
                        <div align="center"><b>Service Difficulty Report</b></div>
                      </td>
                    </tr>
                    <tr>
                      <td>
                        <div align="center"><font size="2"><b>AERONAUTICAL
EQUIPMENT</b></font></div>
                      </td>
                    </tr>
                  </table>
                </td>
                <td width="52%">
                  <table width="95%" border="0">
                    <tr>
                      <td>
                        <div align="center">FORM APPROVED OMB No. 2120-0008</div>
                      </td>
                    </tr>
                    <tr>
                      <td>
                        <div align="center"><b>RIS: WS 8070-1</b></div>
                      </td>
                    </tr>
                    <tr>
                      <td>
                        <div align="center">Control No.<br>

```

```

        <input type="text" name="control_no_f" size="20">
    </div>
</td>
</tr>
<tr>
<td>
    <table width="88%" border="0">
        <tr>
            <td width="51%">
                <div align="center">ATA<br>
                <input type="text" name="ata_f" size="6">
                </div>
            </td>
            <td width="49%">
                <p align="center">Code<br>
                <input type="text" name="code_f" size="9">
                </p>
            </td>
        </tr>
    </table>
</td>
</tr>
</table>
</div>
</td>
</tr>
<tr>
<td>
    <table width="100%" border="0">
        <tr>
            <td><font size="4"><b></b></font></td>
        </tr>
    </table>
    <font size="4"><b><u>MAJOR EQUIPMENT IDENTITY</u></b></font><br>
    <table width="100%" border="0">
        <tr>
            <td width="15%">Enter pertinent data</td>
            <td width="20%">MANUFACTURER</td>
            <td width="18%">MODEL/SERIES</td>
            <td width="19%">SERIAL NUMBER</td>
            <td width="28%">N- Aircraft Registration Number</td>
        </tr>
        <tr>
            <td width="15%">AIRCRAFT</td>
            <td width="20%">
                <input type="text" name="aircraft_man" size="20">
            </td>
            <td width="18%">
                <input type="text" name="aircraft_mod" size="20">
            </td>
            <td width="19%">
                <input type="text" name="aircraft_ser" size="20">
            </td>
        </tr>
    </table>

```



```

        <div align="center">STATUS <br>
        <input type="text" name="status" size="15">
        </div>
    </td>
    <td height="22" width="12%">
        <div align="center">CARRIER <br>
        <input type="text" name="carrier" size="10">
        </div>
    </td>
    <td height="22" width="7%">
        <div align="center">ATA <br>
        <input type="text" name="ata_p" size="6">
        </div>
    </td>
    <td height="22" width="20%">
        <div align="center">AIRCRAFT TYPE<br>
        <input type="text" name="aircraft_type" size="20">
        </div>
    </td>
    <td height="22" width="13%">N-<br>
        <input type="text" name="aircraft_reg" size="20">
    </td>
    <td height="22" width="24%">
        <div align="center">CONTROL NO.<br>
        <input type="text" name="control_no_p" size="20">
        </div>
    </td>
</tr>
</table>
</td>
</tr>
<tr>
<td>TEXT<br>
    <textarea name="prob_tx" cols="101" rows="10"></textarea>
</td>
</tr>
<tr>
<td>SPECIFIC PART CAUSING PROBLEM</td>
</tr>
<tr>
<td>
<table width="90%" border="0">
<tr>
<td>
<div align="center">PART NAME <br>
<input type="text" name="part_name" size="25">
</div>
</td>
<td>
<div align="center">MFG PART NUMBER<br>
<input type="text" name="mfg_part_num" size="25">
</div>
</td>
<td>
<div align="center">PART CONDITION<br>
<input type="text" name="part_condition" size="25">

```

```

        </div>
      </td>
    </tr>
  </table>
</td>
</tr>
<tr>
  <td>
    <table width="90%" border="0">
      <tr>
        <td width="52%"><font size="2">COMPONENT?APPLIANCE ABOVE PART
          INSTALLED ON</font><br>
          <input type="text" name="comp_app_inst" size="35">
        </td>
        <td width="6%">Report<br>
          whole<br>
          hours </td>
        <td width="23%">
          <div align="center">PART TT <br>
            <input type="text" name="part_tt" size="15">
          </div>
        </td>
        <td width="19%">
          <div align="center">PART TSO<br>
            <input type="text" name="part_tso" size="15">
          </div>
        </td>
      </tr>
    </table>
  </td>
</tr>
<tr>
  <td>
    <table width="90%" border="0">
      <tr>
        <td width="24%">
          <div align="center">COMP/APPL NAME<br>
            <input type="text" name="comp_appl_name" size="20">
          </div>
        </td>
        <td width="25%">
          <div align="center">MANUFACTURER<br>
            <input type="text" name="manufacturer" size="20">
          </div>
        </td>
        <td width="28%">
          <div align="center">MFG. MODEL/NUMBER<br>
            <input type="text" name="mfg_model_num" size="20">
          </div>
        </td>
        <td width="23%">

```

```

        <div align="center">SERIAL NO.<br>
        <input type="text" name="serial_no" size="20">
        </div>
    </td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
<tr>
<td>
<table width="100%" border="0">
<tr>
<td><u><b></b></u></td>
</tr>
</table>
<u><b><font size="4">SUBMIT BY</font></b></u><br>
<table width="100%" border="0">
<tr>
<td>
<table width="90%" border="0">
<tr>
<td width="28%">SUBMITTER (check one)<br>
<input type="text" name="submitter" size="25">
</td>
<td width="32%">
<table width="60%" border="0" height="86">
<tr>
<td width="16%">
<div align="center">A</div>
</td>
<td width="15%">
<div align="center">B</div>
</td>
<td width="5%">
<div align="center">C</div>
</td>
<td width="11%">
<div align="center">D</div>
</td>
<td width="10%">
<div align="center">E</div>
</td>
<td width="10%">
<div align="center">F</div>
</td>
<td width="10%">
<div align="center">G</div>
</td>
<td width="13%">
<div align="center">H</div>
</td>
<td width="10%">
<div align="center">I</div>

```

```

</td>
</tr>
<tr>
<td height="70" width="16%">
  <div align="center"><font size="1">CARRIER<br>
    <br>
    <input type="checkbox" name="carrier_ch" value="checkbox">
  </font></div>
</td>
<td height="70" width="15%">
  <div align="center"><font size="1">REP<br>
    STA <br>
    <br>
    <input type="checkbox" name="rep_sta_ch" value="checkbox">
  </font> </div>
</td>
<td height="70" width="5%">
  <div align="center"><font size="1">OPER<br>
    <br>
    <input type="checkbox" name="oper_ch" value="checkbox">
  </font></div>
</td>
<td height="70" width="11%">
  <div align="center"><font size="1">MECH<br>
    <br>
    <input type="checkbox" name="mech_ch" value="checkbox">
  </font></div>
</td>
<td height="70" width="10%">
  <div align="center"><font size="1">AIR <br>
    T AXI<br>
    <br>
    <input type="checkbox" name="air_taxi_ch" value="checkbox">
  </font><br>
</div>
</td>
<td height="70" width="10%">
  <div align="center"><font size="1">MFG<br>
    <br>
    <input type="checkbox" name="mfg_ch" value="checkbox">
  </font></div>
</td>
<td height="70" width="10%">
  <div align="center"><font size="1">FAA<br>
    <br>
    <input type="checkbox" name="faa_ch" value="checkbox">
  </font></div>
</td>
<td height="70" width="13%">
  <div align="center"><font size="1">OTHER<br>
    <br>
    <input type="checkbox" name="other_ch" value="checkbox">
  </font></div>
</td>
<td height="70" width="10%">
  <div align="center"><font size="1">Spec<br>

```



```

        </td>
        <td width="13%">
          <div align="center">SYS.<br>
            <input type="text" name="sys1" size="15">
          </div>
        </td>
        <td width="17%">
          <div align="center">SYS. <br>
            <input type="text" name="sys2" size="15">
          </div>
        </td>
      </tr>
    </table>
  </td>
</tr>
<tr>
<td>
  <textarea name="comments_tx" cols="101" rows="10"></textarea>
  <br>
  ADDITONAL COMMENTS</td>
</tr>
</table>
</td>
</tr>
</table>
<p>&nbsp;</p></div>
<input type="button" value="Submit" onClick="XMLForm(this.form)">
<input type="reset" name="XFormC" value="Clear Information"><br>
</form>

<form name="XAdd">

<textarea name="write" rows=15 cols=80 wrap="phisical">
<?xml version="1.0" standalone="yes"?>
</textarea><br>
<input type="reset" name="XwriteC" value="Clear text area">
</form>

</body>
</html>

```