Automated Data Translation for Avionics Health Management through Intelligent Evidence Streams and Databases

Patrick W. Kalgren
Carl S. Byington, P.E.
Bryan P. Donovan
Impact Technologies, LLC
2571 Park Center Blvd.
State College, PA 16801
814-861-6273
patrick.kalgren@impact-tek.com

Abstract– *The authors have developed automated data translation and storage to support improved avionics health management through intelligent evidence streams and databases. The architecture permits capture of available evidence in legacy systems to exploit interdependencies and relationships of multiple disparate line replaceable units (LRUs) to provide embedded onboard and improved at wing reasoning. This can dramatically increase the effectiveness of current repair and maintenance efforts through a reduction in ambiguity and subsequent decrease in the high percentages of cannot verify (CNV) test results that plague the present system. Evidence and context preservation, demonstrated through intelligent data stream and database design, supports continuous learning and system evolution through technology changes and system upgrades. These developments support an avionics health management system that maintains information continuity from onboard and at-wing to the operational and logistic chain. Higher levels of reasoning and data exploitation at each point offer continued assessment, improvement, and evolution of the system.*

INTRODUCTION

Avionics systems in use today employ components and technologies that span many decades of development. This is a result of substantial evolutionary gains in electronics and the extension of airframe lifespan. The community defines two major classes of avionics systems: (1) federated systems containing many possible technological contrasts within one system and (2) modern integrated modular architecture (IMA) systems that incorporate the highest level of component integration that modern technology can support. Most current avionic systems utilize a federated architecture. Each line replaceable unit (LRU) is an independent device made by different manufacturers using potentially very different design approaches. Technological diversity and fractal design present a host of challenges to the avionics maintenance and logistics process. Each LRU manufacturer provides independent diagnostic capability for its unit in the form of built-in-test (BIT), automated test equipment (ATE), and test program sets (TPS). Non-uniformity of test equipment and unrealized overlap of functional capability results in excess test resources at all levels of the maintenance system and inhibits interoperability through the inflexibility of process. Commonly lost in this process is the working requirement that these disparate avionics components function side by side, in a largely autonomous fashion, to provide the total system functionality required to fulfill the aircraft's mission. It is this integration and its potential system-level effects that have not been considered by the current maintenance infrastructure. Exposing this integration of avionics components through the capture and meaningful retention of all available data can contribute significant intelligence to avionics diagnostics and repair. Previously published works by the authors [1, 2] describe the development of the high level data collection and the application of reasoning techniques. This paper extends the previous work and discusses the automated capture, open systems architecture (OSA) representation, and handling of data available from existing federated systems.

Full utilization of available at-wing, ongoing avionics operational, and diagnostic information requires the maintenance of data continuity through preservation of context and assurance of data integrity. These concerns necessitate a solid data translation and storage infrastructure that supports automation at every data transfer point. Translation of LRU built-in-test (BIT) data along with flight data, environmental conditions, and other available information into intelligent, open data streams enables application of reasoning techniques throughout the maintenance process, supports automated data storage in networked databases, and permits data mining and trend development of asset failures and repairs. Automated, event-driven processes can replace hand written flight notes, maintenance action forms, and LRU error sheets throughout the organizational ('O'), intermediate ('I'), and depot ('D') facilities, extending the availability of important data to decision making processes. Automated evidence transport to each subsequent level of maintenance ensures the integrity of data that has traditionally been entered manually – if captured at all. Evidence availability can enhance diagnostics at each level through application of advanced

Presented at the American Helicopter Society 60th Annual Forum, Baltimore, MD, June 7-10,

reasoning techniques, but also through the preservation of system level context lost in current practices.

"INTELLIGENT" DATA STREAMS

The use of the term "intelligent", with respect to data streams, invites the imagination of many possibilities. While the ideal semantic notion of maximum interoperability and complete machine autonomy is a worthy objective, many levels of acceptable and useful openness exist that are achievable now. Currently available eXtensible Markup Language (XML) schema-based specifications support a level of interoperability and openness not previously possible. Emerging standards will extend this interoperability to include test requirements, system diagnostic models, and even reasoning models to enhance the viability of multiple source knowledge fusion within an open architecture. The ability of XML to represent data and context through meta-data combined with verification and validation using schema specifications provides a powerful tool in the quest for open systems and interoperability. The concept presented here is one in which maximal process automation is achieved in an open-systems environment. Supporting technologies include eXtensible Markup Language, open database schemata, pure data object models, and advanced middleware design. In a system where attainment of a complete, formal, semantic, description is a challenging and prohibitive objective, much of the intelligence with respect to data translation and representation depends on faithful implementation of middleware. Translators that convert proprietary data available at system or component level interfaces to standardized data streams are the first and best opportunity for the accurate capture and preservation of the evidence and context required for the robust application of reasoning at every maintenance level. Openness and accessibility at each interface and step of the maintenance and reasoning process is essential to the successful deployment of advanced, broadly utilized, reasoning systems.

AHM DESIGN CONCEPT

The open Avionics Health Management (AHM) architecture shown in Figure 1 incorporates the capture of all available evidence within current systems, permits the inclusion of future system enhancements, and enables third party evidence sources as well as reasoning modules. Data flow from evidence sources to low-level reasoning is supported by middleware transformation of proprietary data to an Open-Systems Architecture (OSA). An OSA data broker facilitates automated data storage and retrieval and reasoning processes. An OSA knowledge broker supports the system level fusion of reasoning outputs. This picture, from data collection to high-level reasoning and continuous learning, represents a recursive model that incorporates the further emergence of evidence throughout the maintenance infrastructure.

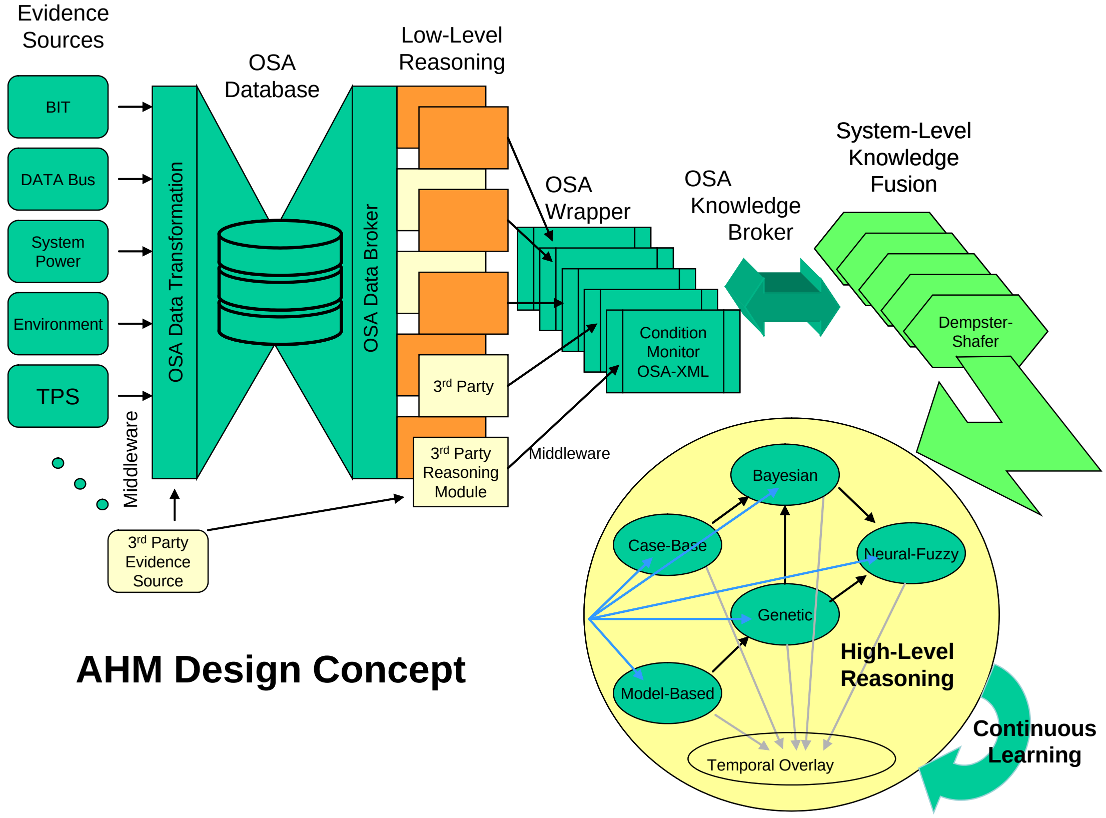

Figure 1 - Avionics Evidence Collection and Reasoning

The overall architecture supports the continual learning, system assessment, and discovery of hidden patterns at each level. Discovered knowledge manifests the greatest possible usefulness at the logistics level where data from individual units, installed in individual systems, merges with other units installed on systems from other aircraft, organizational units, and service branches for macro level mining.

More effective avionics troubleshooting and diagnosis begins onboard with a system level perspective [1, 2] and greater capture and utilization of available data. Time stamp, system power parameters, temperature, vibration, and BIT error code, design logic of BIT errors, reliability, life history, and logistics are a few examples of useful evidence sources (Figure 2).

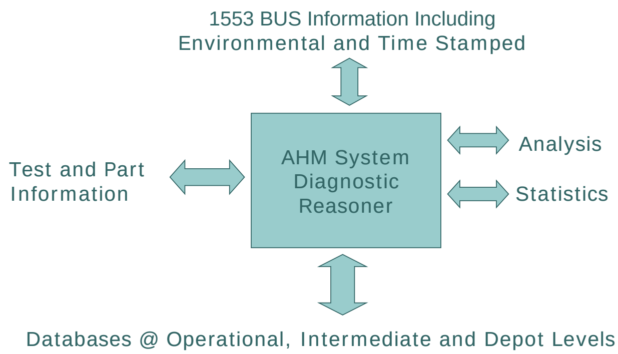

Figure 2 - Reasoning Inputs

DATA TRANSFORMATION

Throughout the maintenance process, automatic capture and conversion of the proprietary data to XML based records adds the supporting meta-data that preserves context. Schema-based XML meta-tags support open data availability to external reasoning and support the automatic storage of accumulated evidence and diagnostic transactions in an open database. Open data architecture enables this type of data transformation. Although openness usually refers to free and unconstrained sharing of information, "open systems" design facilitates the integration and interchangeability of components from a variety of sources. Utilizing open data architecture ensures that third party information can be provided by component/software suppliers to all aspects of the system as seen in Figure 1. Open data architecture also facilitates data continuity throughout the onboard and off-board system by encapsulating a standard representation of information (system id, sensor/LRU/SRU outputs, data collection parameters, etc) represented by the data stream. A standardized meta-data description enables context preservation. Data streams of the XML type are useful due to the illustrative mapping of meta-data (header tags) to database fields. Throughout this paper the XML based data streams are discussed, but the reader should keep in mind other types do exist. Careful information system design necessitates the use of data modeling approaches. A well designed data model provides portability across instantiation types.

Choices for implementation of open communication standards utilizing XML range from the well developed, Open-Systems Architecture for Condition-Based Maintenance (OSA-CBM) to the emerging Automated Test Markup Language (ATML) for avionics test equipment. Other relevant implementations exist, such as Predictive Model Markup Language (PMML), developed for data mining approaches. These existing and emerging standards provide good interim and long term solutions to various aspects of the open architecture that, once implemented, will provide a solid foundation and can be readily migrated toward a mature implementation. Brief descriptions available of technologies are presented here.

OSA-CBM

OSA-CBM is an open system standard developed for machinery health management that consists of a seven-layer architecture based on a collection of similar tasks or function. OSA-CBM is designed around a pure object data model and examples are available instantiated in XML, COM/DCOM, or CORBA forms. The hierarchical levels represent flow of information from the lowest level, sensor output, to the upper most levels of decision making and presentation. The information flow shows a logical progression throughout the layers.

The OSA-CBM layered architecture places no restrictions on inter-layer communication. Non-adjacent layers can communicate with one another. However, constraints could be enforced on an application basis. Figure 3 is a breakdown of the OSA-CBM layers.

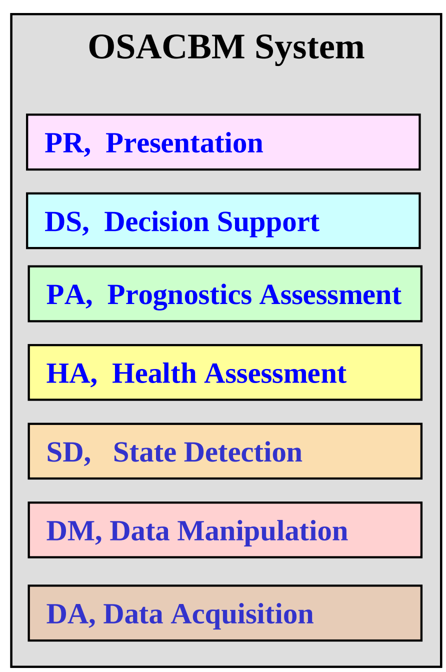

Figure 3 - OSA-CBM Functional Layers

OSA-CBM architecture incorporates communication standards for communicating between layers and modules.

Modalities described by OSA-CBM include XML, CORBA, and COM/DCOM.

To create XML output that conforms to a schema, data wrappers employ handlers. Handlers are program modules that enable schema cross validation. Wrappers use handlers to interpret the incoming data stream and provide the XML tags and appropriate structure. A wrapper implementation requires no change to the existing software or underlying data, but provides the output standardization that preserves situational context and supports information continuity. The design goal of XML schemata, whether ATML, OSA-CBM, or other implementations, is to represent data streams in a manner that preserves context and provide access to data in an open manner suitable to the communications process at hand. An additional outcome of schema-based, hierarchical data representation is a natural organization of the data for representation in a database. Context preservation, automated processing, and compatible storage design provide information continuity throughout the AHM system.

EXAMPLE: OSA-CBM DATA EXTRACTION AND CONVERSION OF MIL – 1553 INTERFACE

The authors have demonstrated the collection of proprietary manufacturer BIT information and its transformation to a transportable data stream using hardware in the loop (see Figure 4). A C-130 automatic flight control computer (AFCP), supplied by Honeywell, connected to a laptop computer via a MIL-1553 interface, provided by Ballard Technologies, formed the core equipment demonstration.

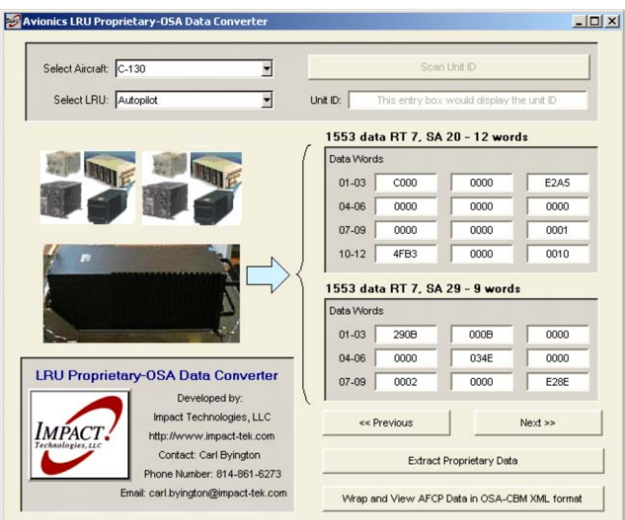

Figure 4 - Impact 1553 Data Extraction and XML Translation Module

An OSA wrapper, developed by Impact Technologies, LLC receives aircraft flight control processor (AFCP) data on the 1553 data bus. A middleware module, developed by Impact, converts the raw, hexadecimal data to an XML data stream which conforms to a condition monitor output specification defined by the OSA-CBM protocol version 1.0.2. The OSA-CBM XML schema provides an open system representation of the data. The CM module best represents the data received from the AFCP built in test, an AFCP "condition monitor". A timestamp added to the beginning of the file marks the time of data acquisition. The open data package is now ready for use by other diagnostic system layers. These layers may include at-wing reasoning applications to aid in troubleshooting or may simply be storage of a repair instance data package for use by downstream test procedures. Innovative programs such as Boeing's Smart TPS [5] seek to incorporate aircraft on-board data into test stand procedures.

AUTOMATED TEST MARKUP LANGUAGE – ATML

The ability of the OSA-CBM standard to represent data requirements at intermediate and depot levels becomes limited and possibly constricting. The ATML working group is developing open standards to describe all aspects of test and maintenance of avionics systems. The core effort targets existing test platforms and the description of required data elements for test configuration, stimulation signals, and the capture of results. The ATML working group is exploring options for the description and modeling of systems and diagnostic reasoning processes. While much has been successfully accomplished, a considerable body of work remains to be completed. Current plans include the exploration and utilization of existing standards to broaden coverage through reuse of work performed in other communities. ATML may ultimately encompass and adopt available standards and provide a uniform resource for data representation at all levels of maintenance.

LEADING OPEN DATABASE CANDIDATES

Avionics information, transformed into a XML stream, is now ready for transport, utilization by reasoning, and database storage. While several possible database architectures exist, such as Oracle, MySQL, PostGreSQL, and TeraData, relatively few open relational schemata have been developed that fit the problem well. The MIMOSA architecture, specifically targeted for machine maintenance, corresponds tightly with OSA-CBM, and provides a good example for a specific relational data storage instantiation. This instantiation, however, has applicability limitations. We present an overview and suggest emerging technologies that may accommodate existing data stores as well.

MIMOSA OSA-EAI

The Machinery Information Management Open Systems Alliance (MIMOSA) has developed "Open System Architecture for Enterprise Application Integration" (OSA-EAI). MIMOSA is a trade association composed of industrial asset management system providers and industrial asset end-users. They develop information integration specifications to enable open, integrated solutions for managing complex high-value assets. The adoption of MIMOSA OSA-EAI specifications facilitates the integration of asset management information, provides a freedom to choose from broader selection of software applications, and saves money by reducing integration and software maintenance costs. Taken as a whole, maintenance and reliability information for aircraft systems is complex. A collaborative maintenance network must provide for the open exchange of equipment asset related information between condition assessment, logistics, and maintenance information systems. The condition assessment sector must include the specialized data required by aircraft id tags, flight operations, environmental conditions, and many other technologies.

MIMOSA provides a standard set of asset management terminology in the MIMOSA Terminology Dictionary. In addition to this document, software designers familiar with Unified Modeling Language can utilize the MIMOSA Conceptual Object Model, written in UML.

MIMOSA OSA-EAI specification is built upon a common information schema allowing information from many systems to be communicated and integrated. The schema is in a relationship form and is known as CRIS-Common Relational Information Schema. CRIS contains standard site, asset, and functional service segment identification nomenclature. In addition, it provides for a method of standard measurement location identification across various condition monitoring technologies. Trendable, scalar data such as operational temperature, pressures and loads are modeled in CRIS. CRIS also allows the communication of diagnostic, health, and prognostic information from smart systems and eases the generation of advisory recommendations. Special maintenance and reliability tables define fields for events (actual, hypothesized, or proposed), health and estimated asset life assessment, and recommendation. CRIS models maintenance and production work request scheduling and the tracking of the completion (or non-completion) of a maintenance or production job as related to an asset. CRIS V2.2 is the latest specification and is published in PDF document form, XML Schema (XSD) form, and in HTML form. In addition to CRIS, MIMOSA experts have generated a large reference database, the CRIS Reference Database Specification in XML and HTML form. Version 2.2 of this database contains many useful codes allowing standardization across many disparate systems – even those from various countries. Version 3.0, tentatively scheduled for release in August 2004, offers additional attractive modifications that will continue to permit the MIMOSA standard to evolve.

Advantages provided by the MIMOSA architecture merit consideration even when the choice is made to develop or use another schema. Open XML standards provide low cost connectivity. A unified asset registry reduces potential errors due to field re-entry. Automation of manual data entry provides increased accuracy. These summarize some of the important advantages provided by a well designed schema.

AVAILABLE TOOLS FOR XML TO DATABASE TRANSLATION

Automated database design provides an answer to development of data storage solutions at localized points of interest. The hierarchical structure of XML documents provides a good starting point for relational schema auto-generation. Enabling technologies are emerging from ongoing research to perform efficient automated relational schema design directly from XML document schema. However, of greater long term use are tools that facilitate the automated insertion of data stream elements into existing storage structures. Automated transport tools and techniques are available now, in commercial form, to facilitate the insertion of XML document data into databases. XML documents fall into two broad categories: data-centric and document-centric. Data-centric documents are those where XML is used as a data transport. To store and retrieve the data in data-centric documents, software is required to convert from the XML format to the database format. An 'XML-enabled' database, is tuned for data storage from XML streams. Usually built into the database or enabled by third-party software, such as *middleware* or an *XML server*, the middleware maps the XML document schema (DTD, XML Schemas, RELAX NG, etc.) to the database schema. The data transfer software is then built on top of this mapping. One software technique uses an XML query language (such as XPath, XQuery, or a proprietary language) or simply transfers data according to the mapping (the XML equivalent of SELECT * FROM Table). Two examples of middleware are Attunity Connect and DB2XML. Attunity Connect, developed by Attunity Ltd, is a heterogeneous query engine that supports bi-directional access to a large number of databases, both relational and non-relational. XML schemata directly support automated database population. Middleware parses incoming OSA-CBM data streams and directs data elements to the appropriate tables within the database. The XML meta-tags are no longer required, as the organization of the database now serves the meta-data purpose of context preservation. A reverse flow procedure transports query data to other OSA modules operating on the database. DB2XML, developed by Volker Turau, uses Java classes to transfer data from a relational database to an XML document. These classes operate in a standalone application or as servlets. The product models the XML document as a set of tables, for which the user specifies one or more SELECT statements.

UTILIZING EXISTING DATABASES

Many databases exist in the current logistics and maintenance infrastructure. Utilization of these data sources falls into two broad categories. Storage of captured information and knowledge discovery of related, supporting information. Existing data sources range from HUMS ground station and research databases developed by interest groups such as the Rotorcraft Industry Technical Association (RITA) to logistical databases developed and maintained by the services. Usage of legacy data stores for knowledge discovery present unique problems. However, a complete redesign of existing systems to support new endeavors presents a possibly insurmountable hurdle. Organizational inertia, overwhelming costs, and competing interests represent a few of the most relevant obstacles. Moreover, existing structures may meet original design objectives effectively. Integration of data from multiple, disparate, sources is an active area of research. Some promising technologies are emerging that range from distributed data mining and high-speed transport schemes to mining model open representation.

Storage of captured maintenance process related information into existing structures is a problem with immediate relevance and available solutions. The design of middleware will support the automation of data entry where a manual process may currently exist. Maintenance Action Forms (MAF) serve as an example. Currently, a wide variance of capture and retention of MAF data exists across platforms. While one platform may attempt to retain all maintenance information, another may incorporate processes that are readily circumvented or may be non-existent. Automated processes, enabled by COTS hardware technology and facilitated by XML streams and middleware, will add value to the existing system. A prototype application is discussed next.

Current 'O'-level diagnostic practices are limited. They range in scope from minimal at-wing analysis to complex interactive electronic technical manual (IETM) driven troubleshooting attempts that may prescribe in-depth processes. However, experience indicates that troubleshooting efforts beyond a few independent steps rarely succeed and even the best directed processes often break down into an ad-hoc pull-and-replace mis-adventures. The authors have developed a working example of an at-wing portable maintenance aid (PMA) with embedded reasoning (ReasonPro – At Wing™) that provides an interactive approach to 'O'-level maintenance and diagnostics. Innovative reasoners use available BIT data to provide a system level approach to diagnostics that optimizes the path to ambiguity reduction. The design of technologies that provide accurate diagnoses, support evidence and context preservation, and integrate well into the current practices to make maintenance and logistics tasks more effective is critical to the eventual adoption of the developed solutions. ReasonPro At-Wing, enabled by COTS technology, is an example of a transformation from inadequate existing practices into a process which adds value at every level of the current infrastructure.

A PDA-based graphical user interface enables a technician to interactively query and test multiple LRU BIT systems. Impact chose the Windows Pocket PC platform for the implementation of this portable maintenance aid (PMA). This choice offers a wide choice of platforms to end users and supports XML based communication protocols and database interaction with a wide array of developer solutions. Impact developed ReasonPro – At Wing™ using Microsoft's .NET Compact Framework. The framework contains smart device development platform for the Microsoft .NET initiative and is tailored towards the Pocket PC operating system. The application is built for Microsoft Pocket PC 2003 using the C# classes.

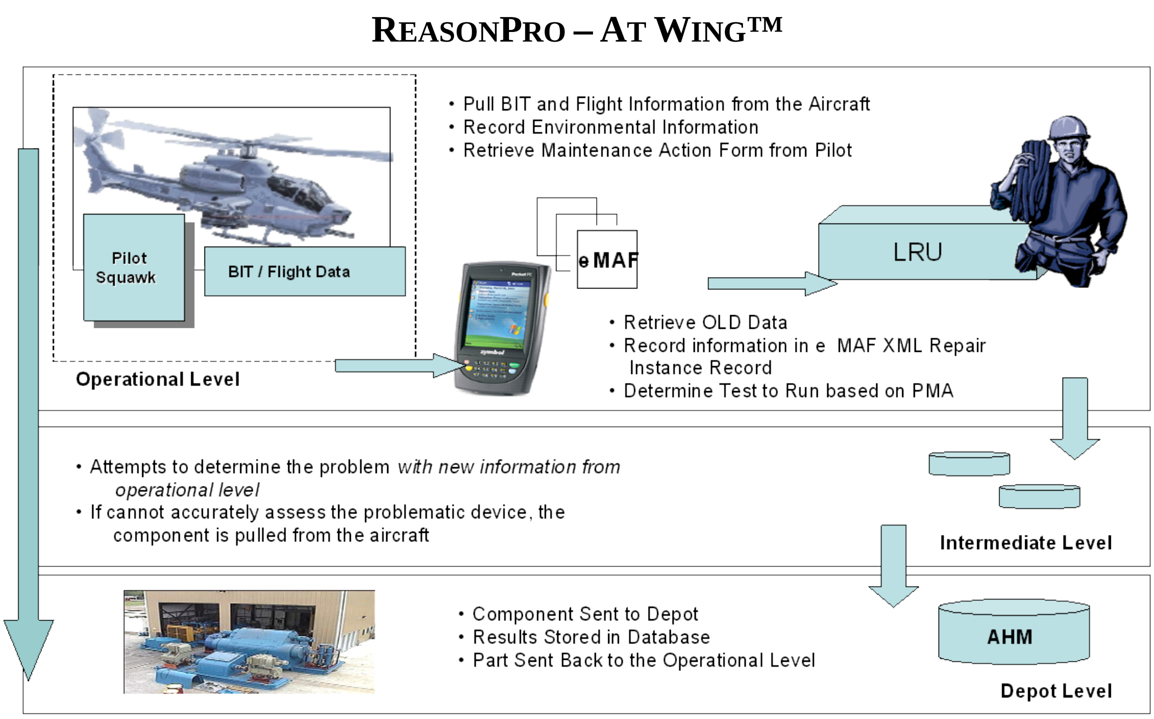

Figure 5 - Impact Avionics Health Management in Practice

The program, powered by Impact's ReasonPro automated reasoning process, assists the maintenance worker in diagnosis of the avionics system. Beyond the immediate value provided by effective ambiguity reduction is the automated capture and retention of on-board and repair instance data.

PROCESS FLOW AND REASONING

The levels of information captured and utilized during a maintenance action (see Figure 5) include aircraft and aircrew inputs, maintainer actions, and command level requirements. The first level contains the squawk data and BIT/Flight data. The second level, the maintainer, receives all information from the first level. ReasonPro – At Wing™, residing at the 'O' level, incorporates the pilot squawk, BIT data, and plane information retrieved from the aircraft. Downloading the problem codes from the plane retrieves BIT/Flight information. This data is transformed into an XML based repair instance record to be later stored in a database. The maintainer then runs the reasoning process of ReasonPro – At Wing™ to determine the maintenance test to run. The application intelligently determines this test based on the information and codes retrieved from the aircraft. The application transforms the data from the tests and aircraft into instances in a local database as an electronic MAF. ReasonPro – At Wing™ then uploads the instances to command level ground databases.

The ReasonPro – At Wing™ application allows the mechanic to easily determine the optimal next diagnostic step and interactively proceed to test LRU's while operating at-wing, within an avionics system. Figure 6 is a series of screen captures that provide a process flow description of the ReasonPro – At Wing™ program. A database on the PDA maintains information essential to the local process and current aircraft configuration. When used with a bar code scanner enabled device, such as the Symbol PDA used for this demonstration, ReasonPro – At Wing™ will capture a unique ID for each serialized component under test. This ID is used to retrieve available configuration and test information from the local database. Air crew reports can be captured and digitized in various forms. ReasonPro – At Wing™ is currently equipped with an innovative BIT reasoner that incorporates system level design knowledge and LRU level BIT data to perform a diagnosis. The reasoning engine uses a positive/negative evidence evaluation to reach an initial ambiguity group. If a clear conclusion is not available, the ambiguity reduction process can link with an Interactive Electronic Technical Manual (IETM) at an optimal entry point and begin system testing. This process iterates as necessary and converges to a rapid solution. The application concludes the root fault through evidence driven diagnostics and interactive ambiguity reduction.

ReasonPro – At Wing™ records the evidence and diagnostic process information digitally in an open XML format and updates ground databases. This information may be valuable to the next levels of LRU diagnostics at the 'O'

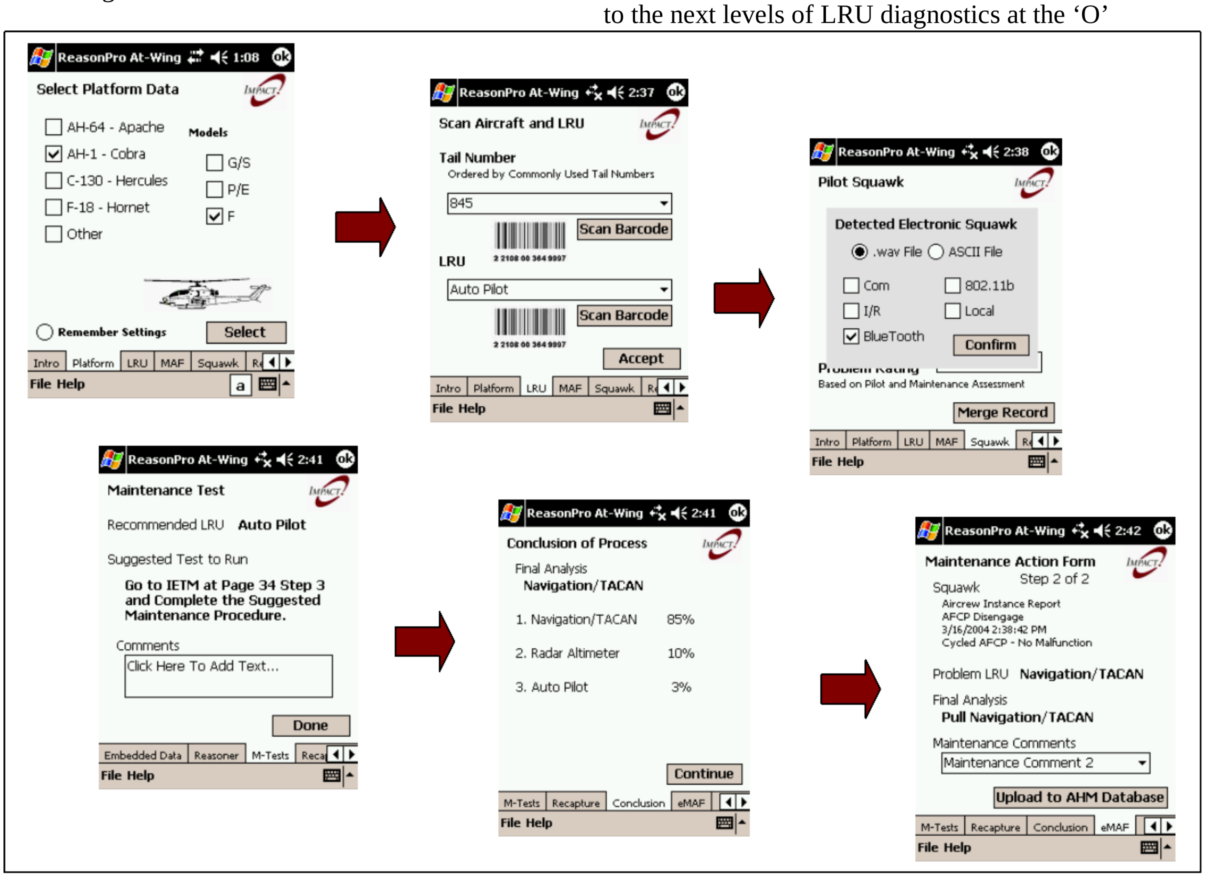

Figure 6 ReasonPro - At Wing™ Diagnostic Process and Data Flow

and ‘D’ maintenance levels. The modular design and the robust reasoning implementation permit ready adaptation to an array of avionics system and supports modifications and facilitation of diagnostic module updates.

REASONER EVOLUTION

A well-designed AHM system should evolve as the experience base becomes broader and as the equipment under management changes. The AHM knowledge discovery level provides continuous learning through data mining and advanced reasoning. Data mining is a technique used on large datasets where relationships may be more complex than simple correlation. Reasoner evolution and AHM design analysis benefit from the implementation of knowledge discovery processes. Knowledge discovery is the process of analyzing large amounts of data to identify or discover subtle differences, patterns, or relationships. Data mining uses statistical analysis, machine learning, modeling techniques, and database technology. The goal is to construct predictive models based on data relationships thereby revealing previously unknown trends in the data.

The process of determining a relationship in a dataset is a multi-phase.

- Exploration and Preprocessing of Data – This phase involves exploring and querying the data to gain insight and also includes data focusing and data validation.
- Modeling – The modeling phase of data mining pertains to selecting an algorithm. The algorithm has three main functions, selecting model representations to fit the data, selecting score functions, and specifying the methods and algorithms to optimize the score function. The model structure is a global summary of the data set and can make a statement about any point in the set. A model structure is a broad view of the data while a pattern structure describes a structure that pertains to only a small portion of the data. Once the model has been established, the model needs to be fitted to determine robustness by using a score function. The score function quantifies the fit between a model structure and given dataset. The score function becomes the means of optimizing the algorithm. The goal of optimization is to determine the structure that achieves the optimal score function.
- Mining – The actual mining of the data is an iterative process whereby the chosen model or algorithm is run on the data set.

After data mining is accomplished, evaluation of the output is an important conclusion to the process. The data owner, to assess the value, usefulness, should review the model and output in terms of what is known about the data. The overall goal of data mining is to extract unknown, useful information from potentially uninteresting data. This is accomplished by summarizing and finding patterns and relationships in the data. Data mining is a multi-step process where data is analyzed and examined with multiple model structures.

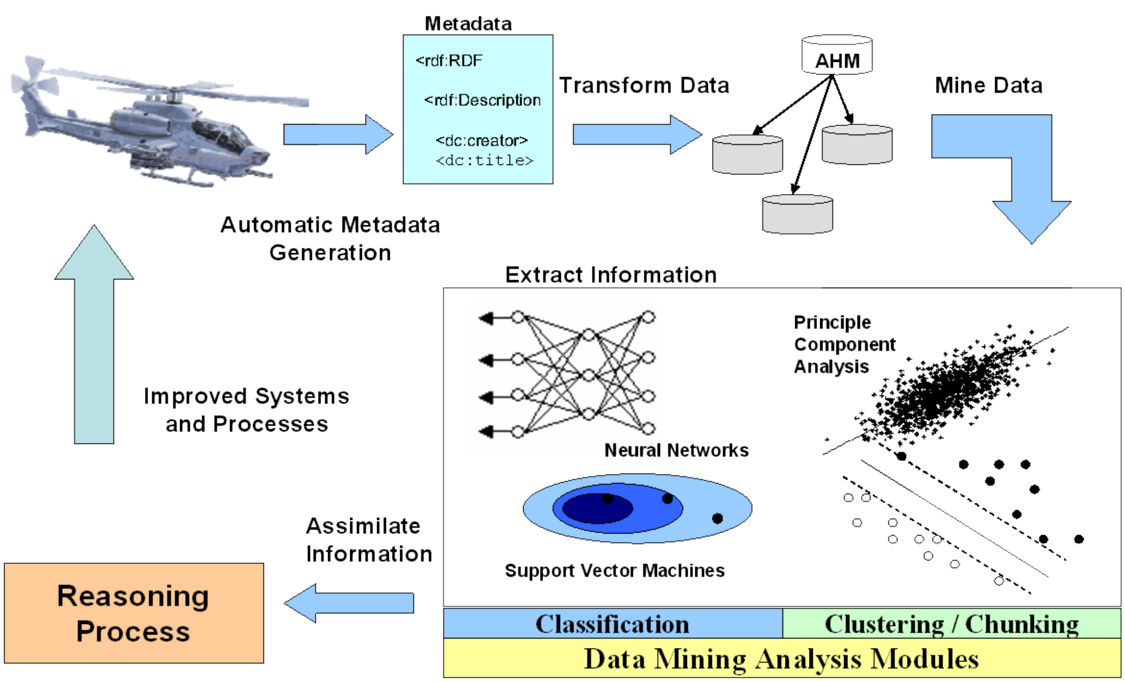

Figure 7 - Data Mining/ Continuous Improvement Cycle

Knowledge Discovery Techniques

Available data mining techniques range from well understood statistical methods to innovative textual-based context analysis. Descriptive and predictive modeling form two large classes of knowledge discovery. There are many forms of descriptive modeling including cluster analysis - probability distribution partitioning of the p-dimensional space into groups and dependency modeling - relationship modeling between variables. Clustering algorithms discover the natural groups in data. However, most data sets do not contain clear, sharp boundaries between data groups.

Principle Component Analysis (PCA) is a form of descriptive modeling. PCA clustering analysis is designed to capture the variance in a dataset. Principle components are the set of variables that define a projection in the data that encapsulates variation. Principle components are orthogonal. PCA reduces dimensionality to summarize the most important parts and filter out noise.

Predictive Modeling seeks to predict the value of a single variable based on known values. The two main categories of predictive modeling are classification and regression modeling. The difference between predictive and descriptive is that prediction has an objective, a predictive value. Descriptive models have no single central variable.

Support Vector Machines are an example of a predictive modeling technique. More importantly, support vector machines are capable of text or language classification and reorganization.

Rule-based Discovery, another technique with text processing capability, draws patterns from the data based on rules. The algorithmic techniques are based on association rules. The rule-based discovery attempts to determine unusual pattern behavior in the context of normal variability.

Future work in AHM system development will include the association of relevant knowledge discovery techniques and related mining parameters with available data stores.

Maintaining Information Openness

Data mining and knowledge discovery efforts can be directly supported through existing and emerging tools by a careful implementation of data stream schemas. Predictive Model Markup Language (PMML) is a tool used widely by the data mining community to support automated knowledge discovery through XML schema driven analysis. This aligns well with the overall XML implementation from real-time push to easily supported databases. PMML supports the open representation of data mining models and associated rules regarding the handling of missing data elements and other configuration items essential to the data mining process.

Conclusions and Future Work

An open architecture to support automated avionics system data collection, advanced reasoning, archival, and knowledge discovery has been described. This system, implemented in a well thought out approach, can provide beneficial enhancements to the current avionics maintenance and repair process without disrupting current practices. A functioning prototype of ReasonPro – At Wing™ has been illustrated. Techniques demonstrated including the use of XML schema-based data streams to preserve situational context and support open communications at various data interfaces encountered within the aircraft maintenance infrastructure. Aspects of this architecture are being employed in emerging systems that seek to improve maintenance of existing and future systems. As the useful lives of existing military platforms extend beyond design expectations and the military migrates to greater interoperability, well designed information systems will enable the transformation of current stove-piped methods into efficient processes that extract the benefit of every piece of knowledge available at each step of the maintenance and logistics cycle. Future work will build upon the solid foundation provided by early attempts at evidence and context capture designed in open frameworks. The gradual elimination of proprietary interfaces will permit the participation of new actors with fresh ideas and novel approaches to knowledge discovery, automated reasoning, and ultimately better maintained and ready military systems. This paradigm shift approach begins with process improvements in the handling of data on existing systems and will point the way to better design and monitoring of next generation systems.

Mr. Patrick W. Kalgren *is a project engineer at Impact Technologies and has a 20-year background in mechanical and electronic system analysis, diagnosis, and repair. He has published technical papers ranging from metrics for HUMS systems and the development of advanced avionics PHM systems to curriculum development for technology education. While employed by PSU ARL, Patrick researched automated classifiers and developed performance tests. At Impact he has lead development of advanced signal processing, applied AI techniques to fault classification, researched advanced database design, and supervised various software projects related to vehicle health management. He is currently leading a program to develop improved diagnostics for avionics maintenance. Patrick has a degree in computer engineering and is a member of Tau Beta Pi, IEEE, and the IEEE Computer Society.*

Carl S. Byington *is a Professional Engineer and the Director of Research and Development at Impact Technologies in State College, PA. He possesses over 15 years in the design and analysis of propulsion, fluid power, thermal, and mechanical systems, and he leads the development of state-of-the-art machinery monitoring and fault detection software and systems for defense and industry applications. In past work at the*

Penn State Applied Research Lab, Carl led teams of engineers and scientists to develop predictive diagnostics algorithms as the Head of the Condition-Based Maintenance Department. He served as the PI on a University Research Initiative for Integrated Predictive Diagnostics, and he subsequently led several programs related to Joint Strike Fighter subsystem prognostics efforts. He has also led helicopter diagnostic algorithm development and fault classification efforts as part of multiple Office of Naval Research programs. Mr. Byington is active in the Machinery Failure Prevention Technology (MFPT) Society. He is also a member, instructor, and past keynote speaker for the Society of Tribologists and Lubrication Engineering society. He serves as the current Chairman of the Machinery Diagnostics and Prognostics Committee within the ASME Tribology Division. Carl has degrees in mechanical and aeronautical engineering, and he has published over 55 publications related to machinery prognostics and health management technologies.

REFERENCES

[1] Byington, Kalgren, et. al., "Embedded Diagnostic/ Prognostic Reasoning and Information Continuity for Improved Avionics Maintenance," IEEE AutoTestCon Conference Proceedings, September 2003.

[2] Byington, Kalgren, et. al., "Advanced Diagnostic/Prognostic Reasoning and Evidence Transformation Techniques for Improved Avionics Maintenance," 2004 IEEE Aerospace Conference, March 6-13, 2004, Big Sky, MT.

[3] Borne, R.O., "Intelligent Embedded Diagnostics/Prognostics for Future Avionics Systems –An Overview," Naval Air Systems Command

[4] F/A-18 E/F Built-in-Test (BIT) Maturation Process*; web: http://www.dtic.mil/ndia/systems/Bainpaper.pdf*

[5] Wegener, S.A., "Smart Test Program Set (TPS)," IEEE AutoTestCon Conference Proceedings, September 2003

[6] Augustin, M., et. al., "eRotor: A Windows CE Portable Flight Management Computer (FMC)," Proceedings of the AHS 59th Annual Forum, May 6-8, 2003, Phoenix, AZ

[7] OSA/CBM Website, http://www.osacbm.org/

[8] Arciniegas, F.A., "Design Patterns in XML Applications," O'Reilly xml.com, Jan. 2000

[9] Mitchell, J S, "MIMOSA - Building the foundation for 21st Century optimized asset management," Sound and Vibrations, September, pp. 12-19. 1995. http://www.mimosa.org/

[11] Predictive Failure and Advanced Diagnostics for Legacy Aircraft: Review of Electrical and Electronics Systems, Prepared for the AFRL, Logistics Sustainment Branch, 2001

[12] Bennett, Douglas, "Regulatory Changes in Avionics Maintenance," Avionics Test Equipment Handbook

[13] Data Mining Group, PMML Documentation, www.dmg.org, 2003