A Health Assessment Database System Implementation for Integrated System Health Management

Mark Turowski, Dr. John Schmalzel - Electrical and Computer Engineering Department
Dr. Adrian Rusu, Ali Daneshmand, Peter Flanner - Computer Science Department
Rowan University – Glassboro, NJ

Keywords – Integrated System Health Management, Health Assessment, Rocket Test Stand, Condition-based Maintenance, Relational Database, Anomaly Detection, IEEE 1451, Smart Sensors

Introduction

Integrated System Health Management (ISHM) is becoming increasingly important to the future of aerospace, military and manufacturing operations and maintenance. A robust ISHM architecture can improve performance and safety while reducing the costs of maintaining complex, mission-critical systems. Utilizing distributed smart sensor technology, model-based reasoning and advanced visualization techniques, an ISHM-enabled system presents its operators with a complete picture of the system state and the health of the system's components and sensors. Achieving this robust Integrated System Health Management Architecture requires the development of many emergent technologies and tools, including a strong support database.

This paper proposes a secure, extensible, and scalable Health Assessment Database System (HADS) for the development, deployment and operation of ISHM technology. The need for this database technology is illustrated in the current literature for ISHM architectures. The United States Navy is currently developing an Open System Architecture for Condition-Based Maintenance systems which requires strict and organized management of system configuration information and historical data [1,2]. Several systems for Intelligent Model-based Prognosis are also being developed which require organization of historical data and system models for dynamic system state approximation [3]. Also, the National Aeronautics and Space Administration is developing ISHM prototypes for Intelligent Vehicle Health Management and Intelligent Rocket Test Facilities which rely heavily on databases to support every level of the ISHM architecture [4,5].

Proposed Approach

The proposed database is being developed in conjunction with NASA's Stennis Space Center to fulfill the requirements of the support databases for the ISHM architecture proposed in [4]. For the purposes of defining the database functionality, the proposed ISHM architecture is summarized in Figure 1 as four main levels of operation. The transducer level employs 1451-compliant smart sensors to acquire sensor readings and transmit calibrated data across a network. The sensor level encompasses the health assessment algorithms which will be used to categorize data anomalies and process faults. The system level utilizes Gensym G2's model-based reasoning to diagnosis the overall health and state of the system. The operator level displays this information to the user in a coherent and efficient way enabling the user to perform the necessary intervention to correct or prevent system faults.

Transducer Level Distributed Real-time Data Acquisition
Sensor Level Health Assessment
System Level Model-based Reasoning
Operator Level Visualization

Figure 1: ISHM Architecture Hierarchy

In order to provide the necessary support to the ISHM architecture, the proposed HADS (see Figure 2) is organized into three main components: the system hierarchy model, firmware codebase and historical data repository. The system hierarchy model organizes the data for each system element into a relational database which organizes these elements in the same manner which they are physically connected. The historical data repository organizes the exemplar data for each sensor and relates this data to information about the origin of the test data and descriptions of any anomalies present. The firmware codebase organizes the information contained within the Transducer Electronics Data Sheets (see IEEE 1451), the code for the health assessment algorithms and the associated electronic datasheets. The firmware codebase also relates this information to the sensors, components and anomaly descriptions to which they apply.

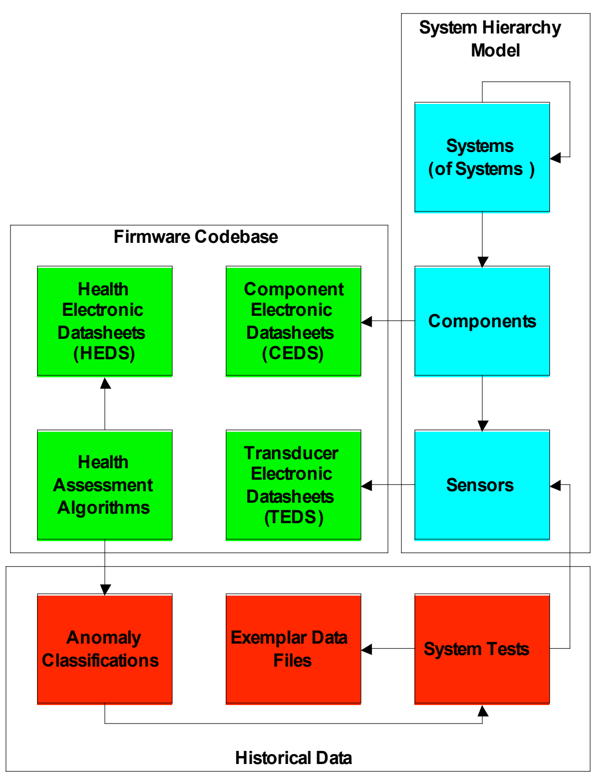

Figure 2: HADS Structure

By combing the historical data repository, the system hierarchy model and the firmware codebase, the HADS provides support for the development, deployment and operation of the ISHM architecture. In the development phase of the project, the historical data repository is used to generate training and testing data sets for the development of health assessment algorithms and HEDS information. The HADS also simplifies the task of organizing the large number of electronic datasheets required for developing a complex ISHM-enabled system. Deploying the system will be greatly simplified as well by the firmware codebase which will provide a centralized location for sensor firmware compiling and deployment. In the completed system, the addition of new anomaly data into the historical data repository will trigger a routine which re-trains the affected health assessment algorithms, recompiles the firmware and remotely update all of the smart sensors. During operation, the HADS will allow the operator to quickly access historical data corresponding to the current system condition as well as provide the model-based reasoning system with the required model information.

Implementation

For development purposes the implementation focuses on the historical data repository. The database has a web-based graphical user interface and was implemented in a **LAMP** [Linux; Apache; MySQL; PHP] environment. Since LAMP is completely open source, the system can be easily and securely set up while providing the necessary features at low cost. MySQL was chosen as the database engine due to its scalability and compatibility with commercial SQL Database systems. Apache was chosen as the HTTP server to provide the graphical user interface for the developer. PHP provides the server-side link between MySQL and Apache. The

Linux Operating System was chosen due to its ease of use for securely integrating MySQL, Apache and PHP. Linux also provides the system administrator with the ability to initiate secure remote connections for its users and strictly control access to the system. The end product, however, will be as portable as possible to promote ease of integration and to ensure a broad sense of accessibility.

In MySQL, highly specific accounts with specific access control have been implemented to ensure that the user cannot access secure data without permission. In order to ensure security, the fields describing the account information are encrypted in order to prevent unauthorized access. In addition to secure logins, each user is assigned a specific access level which is drawn from the user's account information. This set of user account information and access rights are stored in a separate database to allow functionality in the authentication process.

User account information combined with alpha-numeric data from the expert system implies the necessity for an object oriented programming language. This solidifies the need to use PHP in this system. Amongst many features built into PHP are encryption functions, string manipulation capabilities, and a MySQL API. Also, PHP can automatically generate the necessary HTML code for the user interface and use this capability to provide an intuitive web interface for the MySQL data.

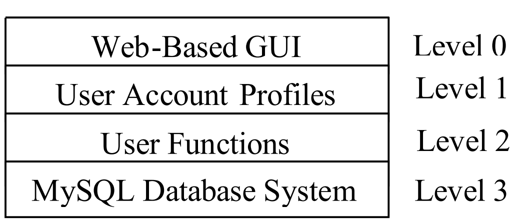

Figure 3: Web Interface Architecture

As shown in Figure 3, there are four general levels to our design. The general premise of this design is that the user enters the system by logging into a secure login algorithm from part of the web-based GUI at Level 0. This login algorithm communicates directly to the database holding the user account information for the sole purpose of authentication. Upon authentication, the user is taken to another part of the GUI that presents the functionality associated with the user's predetermined security level. That functionality is presented to the user through the User Accounts Profiles at Level 1. That serves as the central location in which functionality is defined for each type of user. Level 2 is where the functionality itself is defined. For example, a user with Administrative privileges can perform queries that a Guest user cannot. Finally, Level 3 is the database itself. The database serves as a repository index of the data being held on the server (represented in Figure 4).

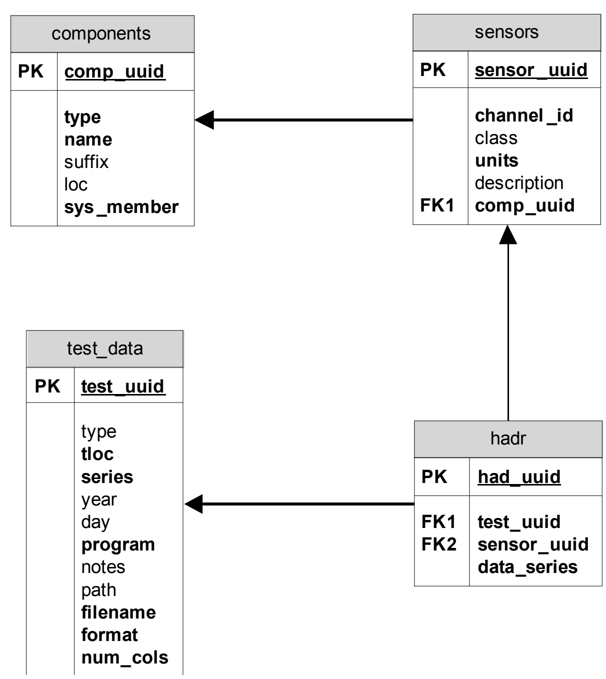

Figure 4: Main Database Design

Conclusion

The historical data repository of the proposed database has been implemented and successfully used to generate training and testing data sets for health assessment algorithm development. This development database is not only providing the developers with a means of quickly sharing and organizing data, but also has substantially reduced the amount of time required to do so. Previously, assembling and sharing a data set of useful size took weeks, and now customized data sets can be drawn from the historical data repository in a few minutes. This has greatly accelerated the process of developing health assessment algorithms and HEDS information. The remaining portion of the system hierarchy and the firmware codebase will be implemented in the coming weeks and linked to the historical data repository to provide the complete Health Assessment Database System.

References

[1] Discenzo, F. M., Nickerson, W., et. al.. "Open Systems Architecture Enables Health Management for Next Generation System Monitoring and Maintenance". http://www.osacbm.org/Documents/General/Whitepaper_osacbm.pdf

[2] Lebold, M. and Thurston, M., "Open Standards for Condition-Based Maintenance and Prognostic Systems," Maintenance and Reliability Conference (MARCON), May 6-9, 2001.

[3] Jianhui Luo, Namburu, M., Pattipati, K., Liu Qiao, Kawamoto, M., Chigusa, S.. *Model-based prognostic techniques.* Proceedings of the 2003 IEEE Systems Readiness Technology Conference (AUTOTESTCON). 22-25 Sept. 2003. Pps. 330 – 340. ISSN: 1080-7725.

[4] Schmalzel, J.; Figueroa, F.; Morris, J.; Mandayam, S.; Polikar, R. *An architecture for intelligent systems based on smart sensors.* IEEE Transactions on Instrumentation and Measurement, Vol.54, Iss.4, Aug. 2005. pp. 1612- 1616

[5] Aaseng, G.B., *Blueprint for an integrated vehicle health management system.* The 20th Digital Avionics Systems Conference (DASC). 14-18 Oct. 2001. Vol. 1. pp. 1-11.

Acknowledgements

- Dr. Fernando Figueroa and the Stennis Space Center ITP Team.
- Dr. Carlos Mata, Becky Oostedyk and the Kennedy Space Center Smart Networked Element Team.
- Dr. Kang Lee and the National Institute of Standards and Technology
- Jon Morris of Jacobs Sverdrup, David Rauth of Kulick & Sofa, and Donald Nickles of Rowan University for their extensive support of this research.