Open Systems Architecture for Condition-based Maintenance (OSA-CBM)

Primer

August 2006

Prepared by

Penn State University / Applied Research Laboratory,
The Boeing Company, and
Machinery Information Management Open Standards Alliance (MIMOSA)

The OSA-CBM specification is a standard architecture for moving information in a condition-based maintenance system. A more in depth look reveals a way to reduce costs, improve interoperability, increase competition, incorporate design changes, and further cooperation in the realm of condition-based maintenance.

The OSA-CBM standard is written in the Unified Modeling Language for software engineers. This primer is intended to cross the gap between computer scientists and program managers and systems integrators. Part 1 discusses how and why OSA-CBM was developed, how it can benefit project managers, and why it should be used. Part 2 discusses the inner workings of the standard and provides a starting place for system integrators to understand and implement the standard.

Part 1: Project Manager's Perspective

What is OSA-CBM?

Project managers implementing condition-based maintenance systems must take on the task of integrating a wide variety of software and hardware components as well as developing a framework for these components. OSA-CBM simplifies this process by specifying a standard architecture and framework for implementing condition-based maintenance systems. It describes the six functional blocks of CBM systems, as well as the interfaces between those blocks. The standard provides a means to integrate many disparate components and eases the process by specifying the inputs and outputs between the components. In short, it describes a standardized information delivery system for condition based monitoring. It describes the information that is moved around and how to move it. It also has built in meta-data to describe the processing that is occurring.

The OSA-CBM v3.1 standard is defined using the Unified Modeling Language (UML) and is designed as a "multi-technological implementation," meaning that it separates the information that can be exchanged in a condition-based maintenance system from the technical interfaces system integrators use to communicate the information. Vendors and integrators can implement the standard using the appropriate technology for their environment. For example, while aircraft health management vendors may elect to use a "real-time" implementation, a vendor developing a portable maintenance aid may elect to implement the standard using XML and web services.

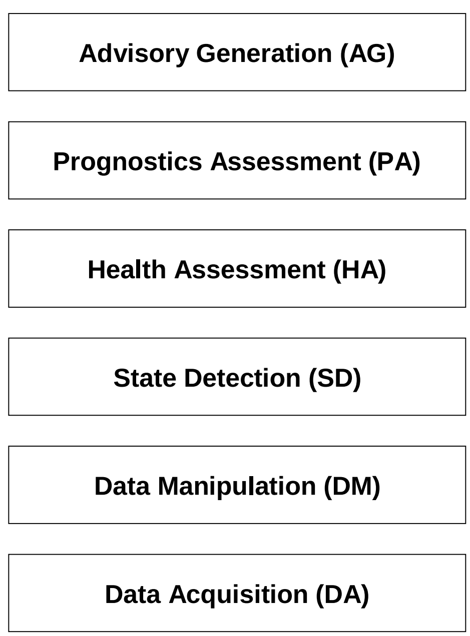

Figure 1 - OSA-CBM Functional Blocks

Who developed it?

OSA-CBM was developed in 2001 by an industry led team partially funded by the Navy through a Dual Use Science and Technology (DUST) program. At the time, no framework or standard existed for implementing CBM systems. The team's participants covered a wide range of industrial, commercial, and military applications of CBM technology: Boeing, Caterpillar, Rockwell Automation, Rockwell Science Center, Newport News Shipbuilding, and Oceana Sensor Technologies. Other team contributors include the Penn State University / Applied Research Laboratory and MIMOSA (Machinery Information Management Open Standards Alliance). MIMOSA is a standards body that manages open information standards for operations and maintenance in manufacturing, fleet, and facility environments.

Why did they develop it?

The U.S. Navy expends billions of dollars every year for maintenance. Most of the costs are in the form of manpower and part of the costs comes from proprietary software and hardware. To control increasing costs the Navy came to industry for ideas, support, and interchangeability standards. The goal was that standardization of information exchange specifications within the community of CBM users would ideally drive the CBM supplier base to produce interchangeable hardware and software components. The team envisioned that a widely adopted open standard would result in a free market for CBM components. There existed a need for an Open System Architecture to facilitate the integration and interchangeability of these components from a variety of sources.

What are the benefits?

Cost – OSA-CBM can provide significant cost savings because system integrators and vendors will not have to spend time creating new or proprietary architectures. Savings will also come from not being committed to single vendors developing entire CBM systems. Since the standard is broken into functional components, multiple vendors may compete to develop select blocks of functionality.

Specialization – When vendors are not constrained to providing an entire CBM system, they can concentrate on one or few areas. The increase in specialization will allow for better algorithms and technology to be developed. Smaller companies that could not provide an entire CBM system can now specialize in one or more of the functional blocks.

Competition – OSA-CBM allows all vendors to use the same input and output interfaces. The separation of functionality from how the information is presented to other applications allows direct comparison of the developed functionalities. Competition now can occur at a functional level, not a system or total solution level.

Cooperation – Not only can competition increase, but cooperation can also. The sectioning of CBM into separate independent blocks will allow multiple vendors to each work on separate modules. Since the standard also defines the interfaces, each module will be able to communicate with the others seamlessly if developed using the same technologies.

DOD Net Centric Architecture – The DoD Net Centric Data Strategy document defines net centricity as, "the realization of a networked environment, including infrastructure, systems, processes, and people, that enables a completely different approach to warfighting and business operations." Net centricity is the networking of many different types of information, including sensors, decision-makers, and support personnel to create a globally interconnected environment where information is readily available. The future vision of the DoD is to operate in a net centric environment where information is shared and utilized by a broad range of applications and decision-makers using a Service Oriented Architecture (SOA). PSU/ARL has shown that OSA-CBM can be implemented as Web Services and comply with the net centric tenets.

What is the relationship between OSA-CBM and MIMOSA?

MIMOSA is the Machinery Information Management Open Standards Alliance (www.mimosa.org). It is a standards body with members from about 50 international companies and some branches of the US Department of Defense. MIMOSA manages and publishes the OSA-CBM standard.

How can OSA-CBM be an open system while still keeping vendor algorithms proprietary?

OSA-CBM is an interface standard and defines the interfaces between the functional blocks in a CBM system. Vendors can develop algorithms to fit inside of these blocks, separating the information processing from how it is presented. This separation allows proprietary code and algorithms to be kept hidden inside each of the functional blocks. It also creates a plug and play capability where vendors can easily insert updates or roll back to previous versions without affecting other modules or programs relying on the functional blocks.

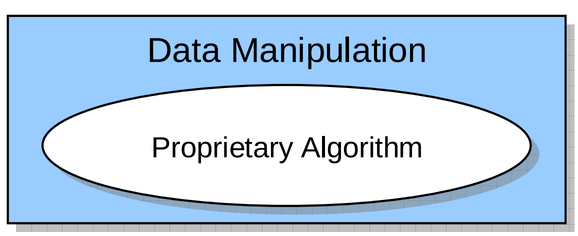

Figure 2 - Example of Proprietary Algorithm in one OSA-CBM Block

How does it relate to ISO-13374 and OSA-EAI?

ISO-13374, Condition Monitory and Diagnostics of Machines, defines the six blocks of functionality in a condition monitoring system, as well as the general inputs and outputs of those six blocks. OSA-CBM is an implementation of the ISO-13374 functional specification. OSA-CBM adds data structures and defines interface methods for the functionality blocks defined by the ISO standard.

The Open Systems Architecture for Enterprise Application Integration (OSA-EAI) is another of the MIMOSA body's family of standards. OSA-EAI defines data structures for storing and moving collective information about all aspects of equipment, including platform health and future capability, into enterprise applications. This includes the physical configuration of platforms as well as reliability, condition, and maintenance of platforms, systems, and subsystems. OSA-CBM uses many of the data elements that are defined by the OSA-EAI. The goal is to have OSA-CBM map totally seamless into OSA-EAI, with future releases.

Can it connect to an archiving system or repository?

The OSA-CBM standard does not include its own data repository. OSA-CBM is designed to store data in an OSA-EAI repository. The majority of data elements in OSA-CBM are derived from those in the OSA-EAI. While OSA-CBM is designed to work with the OSA-EAI, in reality, any archiving system or repository could be used. The specification includes services for data storage but does not specify what type of data source is utilized. OSA-CBM can also be setup to read information from an archive rather than a sensor. This would be useful when processing data that is not being generated in real time.

Can IETMs connect to an OSA-CBM module?

Interactive Electronic Technical Manuals (IETMs) have the ability to connect to external resources to collect error codes. This capability allows the IETMs to display technical information about a problem on the fly. The Health Assessment (HA) module of OSA-CBM can provide errors codes to external applications. IETMs could connect to this service and retrieve the appropriate errors.

Who should use OSA-CBM?

Program managers can specify OSA-CBM in system requirements, vendors can implement OSA-CBM in condition monitoring systems, and systems integrators can use OSA-CBM to integrate condition monitoring and health assessment products from one or more vendors. The standard will save considerable time and money required to develop an architecture and will ease the integration of many disparate systems. The standard will allow PMs to have many vendors develop separate components, not locking them into one vendor and their proprietary software models. This best of breed approach can lead to much better overall systems. OSA-CBM can also be implemented to comply with the DOD net centric tenets and give any CBM system the ability to run in the DODs global information grid.

Small companies can benefit from OSA-CBM by specializing in one or more areas of functionality. They will not have to develop interfaces for each system they want their software to run under, allowing them to concentrate more on the quality of the components. Large companies will have the ability to use modules developed by other companies as part of their overall solution and still remain standards compliant.

Who is already using it?

Boeing is developing a ground analysis system using an OSA-CBM main analyzer. As a vehicle operates, it typically stores many data elements into a sequential store, perhaps binary. Vehicles seldom have a perfect built in test capability, which means improvements may be required to the present systems fault indicators. Changing vehicle operating software is often cost prohibitive or slow.

However, if the right data gets recorded, the systems operational activity can get recorded into a data store. Now this data may be used to determine where design and maintenance improvements can be made. Once it is known how to make the system better, the cheapest and fastest way to get the improvements into the field are via a central design and maintenance site ground-based reasoner. The ground-based reasoner will highlight design improvements or weaknesses or for maintenance issues, filter and improve final system diagnostics and prognostics for the best statement of required maintenance.

Whenever there is a sequential store of data from an operating platform that needs to be run through an analyzer sequentially for a final stage analysis before maintenance requirements are determined, an OSA-CBM based processing system is a good potential choice. This is more applicable if the system is already described in OSA-EAI.

Is OSA-CBM in the DOD Information Standards Registry (DISR)?
Not yet. It is the author's opinion that it will be included in late 2006 or early 2007.

How can Project Managers include OSA-CBM in their requirements?

Project managers can include OSA-CBM in their requirements documents, but need to specify the middleware communication type, such as XML, CORBA, RMI, or DCOM, since it is not specified by the standard. They may also need to specify the type of communication such as synchronous, asynchronous, service, or subscription (described in Figure 4). These choices could be left to the implementer to decide, but multiple vendor systems would require everyone adhere to the same communication technology and type.

Part 2: System Integrator's Perspective

How do I layout an OSA-CBM system?

OSA-CBM consists of 6 independent blocks of functionality which are defined by ISO-13374 (Figure 3). ISO-13374 does not specify how to implement condition monitoring systems, what technologies to use, or what algorithms to implement. Rather, it provides a general framework for the condition monitoring and diagnostics domain.

OSA-CBM defines the data types used for processing and results reporting in a condition monitoring system as well as how that information is moved between process and storage points. This allows each layer or block of functionality to be developed independently. OSA-CBM is an implementation of ISO-13374 and implements the same six functional blocks defined below.

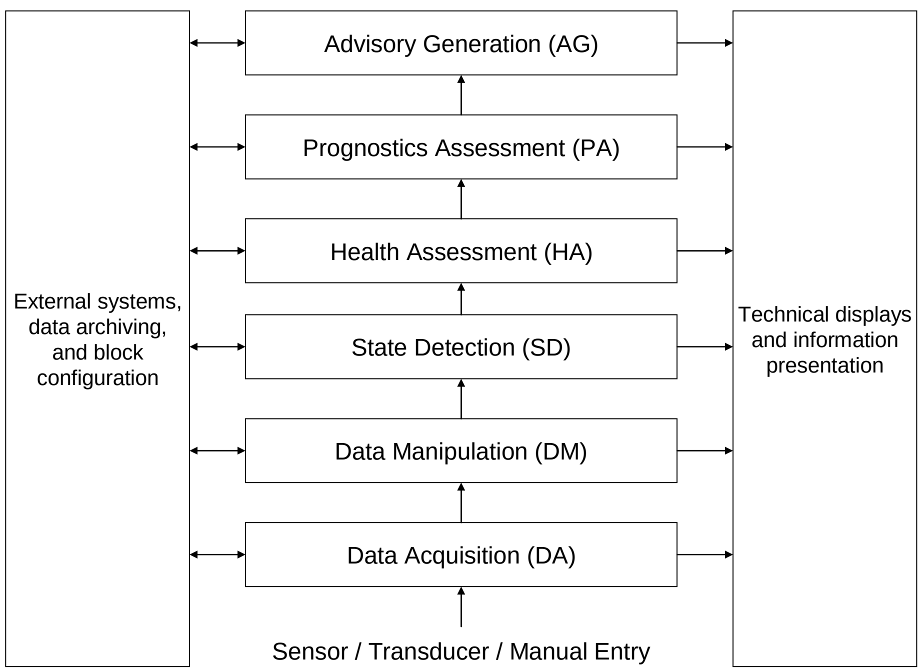

Figure 3 - ISO-13374 Data Processing and Information Flows [6]

The first three blocks are typically technology specific (e.g. vibration monitoring, temperature monitoring, electrochemical monitoring) and provide these functions:

Data Acquisition (DA): converts an output from the transducer to a digital parameter representing a physical quantity and related information (such as the time, calibration, data quality, and data collector utilized, sensor configuration).

Data Manipulation (DM): performs signal analysis, computes meaningful descriptors, and derives virtual sensor readings from the raw measurements.

State Detection (SD): facilitates the creation and maintenance of normal baseline "profiles", searches for abnormalities whenever new data are acquired, and determines in which abnormality zone, if any, the data belong (e.g. alert or alarm).

The second three blocks combine human concepts with monitoring technologies in order to assess the current health of the machine, predict future failures and provide recommended action steps to operations and maintenance personnel:

Health Assessment (HA): diagnoses any faults and rates the current health of the equipment or process, considering all state information.

Prognostics Assessment (PA): determines future health states and failure modes based on the current health assessment and projected usage loads on the equipment and/or process, as well as remaining useful life.

Advisory Generation (AG): provides actionable information regarding maintenance or operational changes required to optimize the life of the process and/or equipment.

What are the major features of the specification?

OSA-CBM blocks provide three types of information: data, configuration, and explanation. *Data* is the information or event sets that the layer has generated. This would be sensor readings for the data acquisition layer and the health of a platform for the health assessment layer. *Configuration* provides information about a module's input sources, descriptions of algorithms used for processing input data, a list of outputs, and various output specifics such as engineering units and thresholds for alerts. *Explanation* is the data or a reference to the data used by a module to produce an output. [4]

Since OSA-CBM is an interface specification, blocks can be implemented separate of each other. OSA-CBM compliant modules can then integrate seamlessly with each other. This allows multiple vendor systems to be easily built and simplifies the process of creating CBM systems.

What are the functions of the OSA-CBM layers?

There are 14 functions that can be implemented by all OSA-CBM layers. Data, configuration, and explanation methods are required. The remaining methods return XML strings that are not specified by the standard. All the methods are listed in Table 1.

Table 1 - Methods for an OSA-Block

Method	Description	Input	Return Type
epRequestDataEvent	Returns a specified DataEvent for the block	MonitorId m	DataEvent

Method	Description	Input	Return Type
epRequestDataEventSet	Returns a specified DataEventSet for the block	MonitorIdList mList	DataEventSet
epGetRequestDataEventSetStatus	Gets the status of a data event request	int RequestID	double
epGetRequestDataEventStatus	Gets the status of a data event request	int RequestID	double
epRequestConfig	Returns a Configuration class that provides the configuration information type – details about the data such as algorithm descriptions, engineering units, assets and segments of the vehicle/plant, agents handling the data, etc…	ConfigRequest configRequest	Configuration
epRequestExplanationDataSet	Provides the actual DataEventSet used for calculations.	MonitorIdList mList	ExplanationDataSet
epRequestExplanationDataRefSet	Provides a handle to a well-known location where the data is stored, such as a database	MonitorIdList mList	ExplanationDataRefSet
epRequestExplanationSrcs	Provides pointers to the data	MonitorIdList mList	ExplanationSrcs
epRequestExplanationSrcsStr	Provides pointers to the data as a string	MonitorIdList mList	ExplanationSrcsStr
epNotifyControl	Allows changing a modules control parameters on the fly	ControlChange controlChange	void
epRequestControl	Returns the modules current control parameters	ControlInfoRequest controlInfoRequest	ControlInfo
epNotifyApp	Sets application specific information	AppRequest appRequest	void
epRequestApp	Returns application specific information	AppRequest appRequest	AppRequestRtn
epRequestErr	Returns error generated by the module	ErrorRequest errorRequest	ErrorNotify

Is OSA-CBM a strict hierarchy?

No. An OSA-CBM block often needs to request information from another block. For example, in a web services implementation one block can add a web reference to the web service providing another block. Although OSA-CBM is often shown as a hierarchical architecture from AG->PA->HA->SD->DM->DA, there is no restriction concerning what blocks can access another. Any block can access any other block, as long as one knows the proper URL or location of the other block. [1]

How do I establish communication between modules?

The OSA-CBM specification defines four types of communication that may be used. They are synchronous, asynchronous, service, and subscription. The OSA-CBM UML defines the interfaces for each module, and the implementer chooses a technology and type of communication to implement the standard. Figure 4 describes the four types of communication.

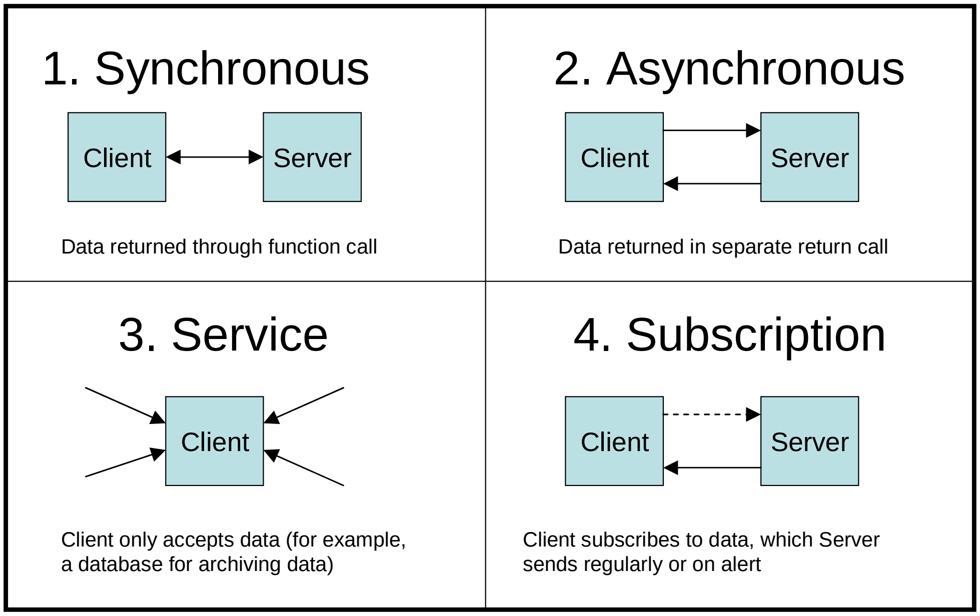

Figure 4 - OSA-CBM Communication Types

What technologies can be used for implementation?

Since OSA-CBM is a multi-technological implementation and defined only in UML, it may be implemented in any middleware programming language such as Web Services, CORBA, Java RMI, DCOM, etc… Two CBM systems developed using different languages will not be able to communicate, but the information they are communicating will be the same. This will make integrating them much simpler because the information and structure will be the same. New interfaces and protocols will not need to be developed.

How can I setup OSA-CBM to work with OSA-EAI?

OSA-CBM and OSA-EAI work in different domains. OSA-CBM is for real time or at time monitoring and processing within a distributed system. OSA-EAI is geared more toward the central or distributed database driven information management activities. OSA-CBM's main job is to be the processing system that develops the best indicator of required maintenance. After this, OSA-EAI gets into its strong domain for maintenance activity management, among other things. OSA-EAI feeds up into the full Enterprise Application Integration level.

OSA-EAI is described by a Common Relational Information Schema (CRIS) which is an XML schema that describes the entities, relationships, and data types for an OSA-EAI data store. OSA-CBM was developed with similar data fields and a similar key structure to OSA-EAI. Currently, approximately 90% of the data elements in OSA-CBM are present in OSA-EAI. Figure 5 shows the areas of information encapsulated by the OSA-

EAI. The outermost green ring defines the three main areas and consists of Open Reliability Management, Open Maintenance Management, and Open Asset Health and Usage Management.

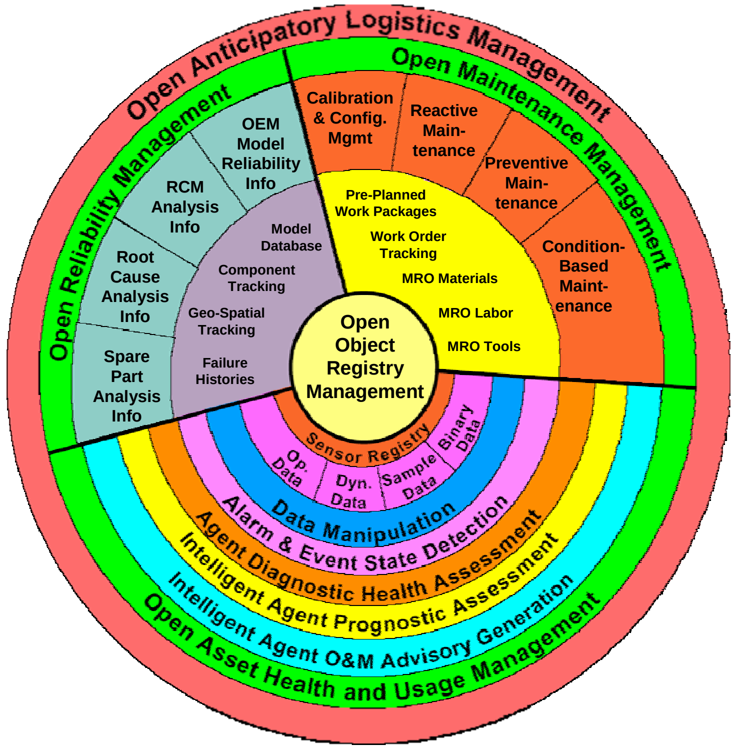

Figure 5 – Major Functions of OSA-EAI

OSA-CBM falls into the Open Asset Health and Usage Management. Data coming from the sensors is the Data Acquisition block and each subsequent block in OSA-CBM is shown as an additional layer on the OSA-EAI figure.

The goal is to have OSA-CBM map totally seamlessly into OSA-EAI. The grand picture is the following.

1. Simple 1-to-1 information component mapping.
2. OSA-CBM extensions to CRIS that have extra information that OSA-EAI does not need.
3. A mapping document where difficult mappings or mappings that have many potential solutions are specified to be done in only one way.

Figure 6 shows an example of how OSA-CBM and OSA-EAI can be integrated using web services.

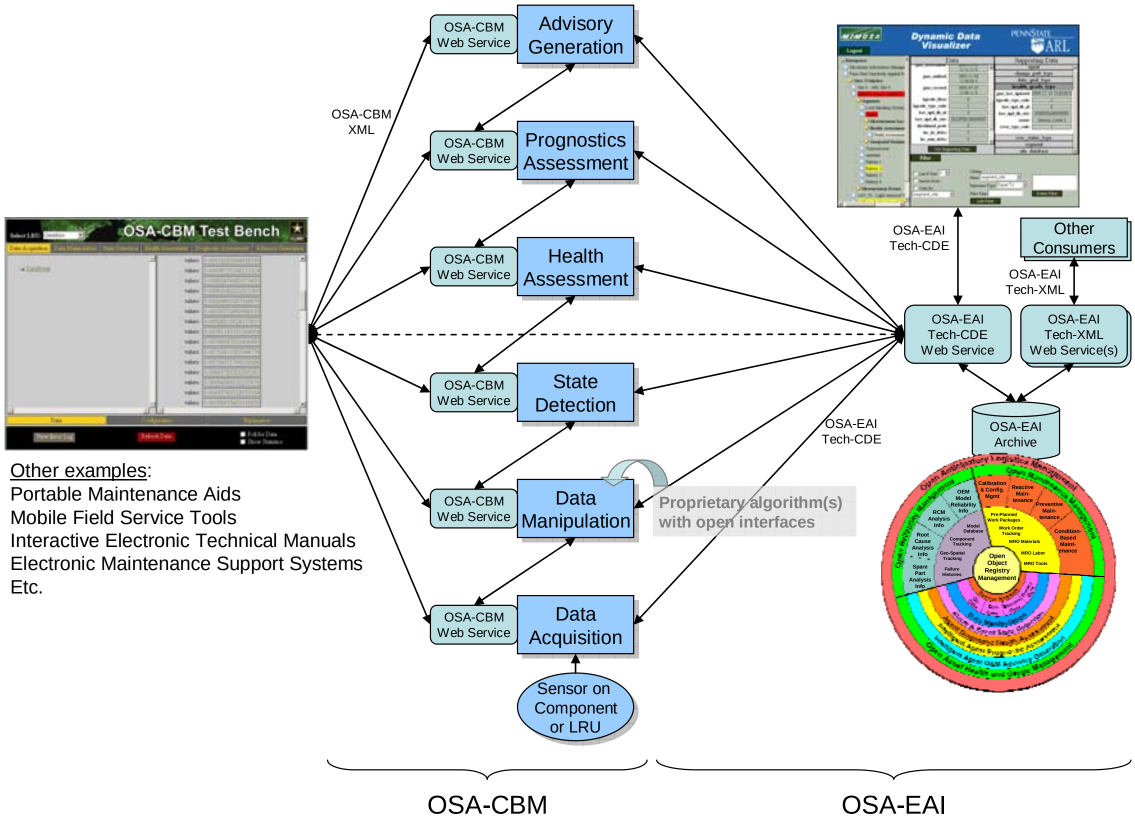

Figure 6 - OSA-CBM Integrated with OSA-EAI

Are there any example implementations and how are they structured?

The Penn State University Applied Research Laboratory did an example implementation of OSA-CBM for the US Army Logistics Transformation Agency (LTA). The implementation was done in XML using Web Services as the middleware technology. Other DOD organizations can request the source code from this implementation and use it as an example or a starting point for integration efforts. The point of contact is James B. Murphy, 717-770-6069.

How is OSA-CBM setup to handle updates to the standard?

The current vision is one of "open to extension, closed to modification". The UML specification will grow as needs arise. Changes or additions to the UML should not produce a ripple effect and change any existing work founded on the base classes. This can only be done through communication within the OSA-CBM subcommittee. When a generic usage need arises it can be discussed, refined and added to the standard. All future projects with a similar need may then implement according to the specified solution.

How would I go about building an OSA-CBM system?

1. Choose a technology for implementation. This must be a middleware type technology such as DCOM, CORBA, Web Services, Java RMI, etc….
2. Turn the OSA-CBM UML into a usable format based on the technology chosen. For example, if Web Services were chosen, the UML would need to be changed to XML.
3. Choose one of the four communication types: synchronous, asynchronous, service, or subscription.
4. Build OSA-CBM compliant interfaces for each block of functionality using the chosen technology and communication types.
5. Build information processing modules for each block of functionality.
6. Put interfaces and info processing modules together as a working system.

Thanks to Bob Walter of Penn State University / Applied Research Laboratory for authoring this document. Bob can be reached at RLW9@PSU.EDU and has offered to field additional questions regarding the standard and include them in future enhancements of this document.

References

[1] Walter R., Lebold M., Boylan D., Puri G., "Data Standard for Processing Transmitting Archiving Prognostic Health Information." *Logistics Transformation Agency and Common Logistics Operating Environment Study*. May, 2005.

[2] Lebold M., Reichard K., Boylan D., "Utilizing DCOM in an Open System Architecture Framework for Machinery Monitoring and Diagnostics."

[3] Pike, C., Ehrensberger, A., "Net Centric Web Services Utilizing a Schema Driven Approach." *Future Combat Systems Logistics Integration Studies*. November 30, 2005.

[4] OSA-CBM v3.1 Standard

[5] Gilbertson D., Dr. Chidambaram, B., Dr. Keller, K., "Condition-Based Monitoring Of an Electro-Hydraulic System Using an Open System Architecture." *Advanced Supportability Concepts Boeing Phantom Works – STL*

[6] ISO-13374 *Condition Monitoring and Diagnostics of Machines — Data Processing, Communication and Presentation*

[7] MIMOSA Web Site: http://www.mimosa.org/

