MIMOSA's Open System Architecture for Enterprise Application Integration (OSA-EAI) Technical Architecture Summary

August 2004

Open Architecture Requirement

Operators, maintenance personnel, logistic managers, OEM's, parts suppliers, and engineers have always wanted to have information about the condition of equipment assets at their fingertips when they need it. Unfortunately, it typically is scattered among separate information systems, one for each platform and then separated by information type: operational data, vibration readings, infrared thermography, oil analysis, control device monitoring, etc.

It is difficult, if not impossible, to view the different information types on the same computer terminal, let alone compile and synchronize them into an integrated view or report on which to base intelligent asset management decisions. Even when the systems can be accessed from the same display, it usually requires separate programs using separate languages.

Interconnectivity of the islands of maintenance and reliability information is embodied in open Enterprise Application Integration (EAI) specifications for Collaborative Maintenance. Previously, these separate information islands were built using specialized proprietary systems that provided value because they were optimized for a specific task or tasks, and they provided best results and value for those purposes. However, their combined value can be multiplied significantly if they can be merged into a Collaborative Maintenance network.

The network can be developed from a collection of information islands by building custom bridges or using an open EAI bridge. The following discussion outlines some of the advantages and limitations of each approach.

Building custom bridges

The Collaborative Maintenance network can be developed by constructing custom software bridges between all the disparate data systems which may be required. This normally involves contracting to an integration company or utilizing in-house information technology resources. However, this route can be very-time consuming and have a very high initial cost and high on-going maintenance costs.

Advantages include:

- High level of customization for commercial and military needs
- High performance tuned to specific bandwidth requirements

Limitations include:

- High risk due to unforeseen incompatibility issues
- High cost due to lack of multiple users
- Difficulty in resolving problems among application suppliers (possible finger-pointing)
- Possible dependence on proprietary interfaces
- High annual software maintenance cost. These costs typically run around 20 percent of original cost which translates to $100,000 annual maintenance costs for a $500,000 software integration project.

Using an open EAI systems bridge

A number of the limitations inherent in custom bridge solutions can be overcome by using an open Enterprise Application Integration (EAI) specification. In mature sectors, this translates into plug-and-play capability that allows a company's information services department to hook up any compliant product to the network.

Advantages include:

- Engineered plug-and-play system capability designed up-front into system
- No burden of on-going integration efforts
- More freedom to choose best technology from information supplier (plug and play)
- Creation of the information backbone of e-maintenance

Limitations include:

- Necessity for suppliers to support industry standard
- Standard gateway may exhibit some performance degradation when custom interface is supplied

Definition of an open system

According to the Software Engineering Institute (SEI) at Carnegie Mellon University (*www.sei.cmu.edu/opensystems/glossary.html)*, a specification is open if its interface is fully defined and available to the public and it is maintained by a group consensus process.

The SEI goes on to define an open system as a collection of interacting software, hardware, and human components:

- Designed to satisfy stated needs
- With interface specifications of its components that are fully defined, available to the public, and maintained according to group consensus
- In which the implementations of the components conform to the interface specifications.

It follows that open system architecture would be made up of components, both hardware and software, that are specified in an open manner.

A number of open EAI consensus-based specifications have been developed or are in development for various information sectors. Those information sectors, specifications, and consensus-building organizations include:

- Control systems and production schedulers: OPC; OPC Foundation (*www.opcfoundation.org*) and ISA—The Instrumentation, Systems, and Automation Society (*www.isa.org*)
- Engineering product data management systems: STEP; Standard for Exchange of Product Model Data, ISO 10303 STEP (*http://cadd.cern.ch/cadd_step.html*)
- Enterprise resource planning (ERP) systems: OAGI; Open Applications Group, Inc. (*www.openapplications.org*)
- Condition monitoring systems: MIMOSA (*www.mimosa.org*)
- Maintenance scheduling (CMMS/EAM) systems: MIMOSA (*www.mimosa.org*)
- Predictive reliability systems: MIMOSA (*www.mimosa.org*)

Operating in close association with MIMOSA is the International Standards Organization's (ISO) Technical Committee 108, Subcommittee 5—Condition Monitoring and Diagnostics of Machines (*www.iso.ch*). Specifications which emerge from this subcommittee are regularly reviewed and adopted into MIMOSA to ease adoption of the standards.

Brief Background on XML

XML is often compared to hypertext markup language (HTML), the document language of the Internet’s World Wide Web.   HTML is a set of rules used to describe how a Web document should look on your computer screen. Tags enclosed by angle brackets in the text of the document file on the server computer indicate how various passages are to be interpreted by the browser on the client computer requesting the page. For example, the tags for the title of this section might be coded as <p><Font size=”+1”>Brief Background on XML</p> for display on a Web page. The tags indicate that the text is a paragraph to be set in boldface type one size larger than the regular font size. This approach works well for documents, but not for structured data.

XML is a set of rules used to create a special markup language for exchanging structured information over the network. It is used to describe both structure and content of the document; how the document should look is handled separately. The rules of XML tagging (mark up) are tightly defined by the World Wide Web Consortium (W3C), but not the language’s structure and vocabulary. Tag vocabulary is defined by a consortium or a corporation and varies according to document content. A MIMOSA asset-related section of an XML document looks like:

```
<asset
   asset_org_site="1367960"
   asset_id="123497423"
   as_type_code="737"
   name31="Motor, AC, Induction"
   user_tag_ident="E3923-271-02"
   mf_duns_code="1367960"
   manuf_trade_name="General Electric Company"
   model_number="E3923"
   revision_number="0"
   serial_number="E3923-271-02"
   gmt_installed="1999-07-01 00:00:00-000000000"
   gmt_removed=""
/>
```

The definition of the XML documents are explained in XML Schema Definition files (XSDs). XML rules ensure that the specialized languages created by XML can be processed on compact computer programs called parsers. Humans can also interpret the XML code. Many of the elements in the previous example taken from the MIMOSA demonstration project are easily recognized without referring to a schema for the data. Further processing is needed to make the information easily read by people.

XML does not include rules for displaying information. Sorting and formatting of the information in the XML document is handled by stylesheet. The stylesheet approach allows a single set of data, such as a work order, to be specially rendered for different devices. The data could be formatted for a desktop computer screen or a palm computer display, or even audible speech.

Advantages of XML

XML provides a standard by which the content structure of a wide variety of information, from simple to complex, can be marked up for easy transfer over the network for use on a variety of computing platforms. XML can be used to mark an ordinary document, a structured record (work order or purchase order), a data record (query result), graphical presentation (user interface), standard schema elements, and more.

Legacy data encoded in XML can be delivered over a local area network or the Internet without having to be changed. Once on the client computer, it can be edited, manipulated, and presented in various views without return trips to the server computer. The widespread adoption of XML should speed up the Internet because more processing could be done locally, reducing bandwidth loads for traffic between the client and server.

MIMOSA's OSA-EAI Specification

The Open System Architecture for Enterprise Application Integration (OSA-EAI) specification has been developed by the **Machinery Information Management Open Systems Alliance (MIMOSA)**. MIMOSA is a trade association composed of industrial asset management system providers and industrial asset end-users which develops information integration specifications to enable open, integrated solutions for managing complex high-value assets.

MIMOSA's OSA-EAI system specifications offer advantages for maintenance and reliability users as well as technology developers and suppliers. For users, the adoption of MIMOSA OSA-EAI specifications facilitates the integration of asset management information, provides a freedom to choose from a broader selection of software applications, and saves money by reducing integration and software maintenance costs.

For technology suppliers, the adoption of MIMOSA OSA-EAI specifications stimulates and broadens the market, allows concentration of resources on core high-value activity rather than low value platform and custom interface requirements, and provides an overall reduction in development costs.

Taken as a whole, maintenance and reliability information is complex. A Collaborative Maintenance network must provide for the open exchange of equipment asset related information between condition assessment, logistic, and maintenance information systems. The condition assessment sector must include the specialized data required by vibration, oil analysis, infrared thermography, motor circuit evaluation, and many other technologies.

Prior to MIMOSA, developers defined data fields to fit their own hardware and software systems. MIMOSA's experts from across the globe have spent more than 5 years in developing the Open System Architecture for Enterprise Application Integration (OSA-EAI) specification (see Figure 1) so that all loosely-coupled systems can exchange information by using MIMOSA's open system architecture. In 2003, the MIMOSA worked with the OPC Foundation to announce OpenO&M™ -- packaging *Tech*-XML data inside OPC's XML-DA Complex Data specifications. This allows OPC-compliant operational HMI display and control systems with need for complex data such as asset registry information, diagnostic/prognostic information, maintenance work tracking /operator action tracking , reliability, and complex data (vibration, image, video, etc.) to retrieve or post this information in an open manner.

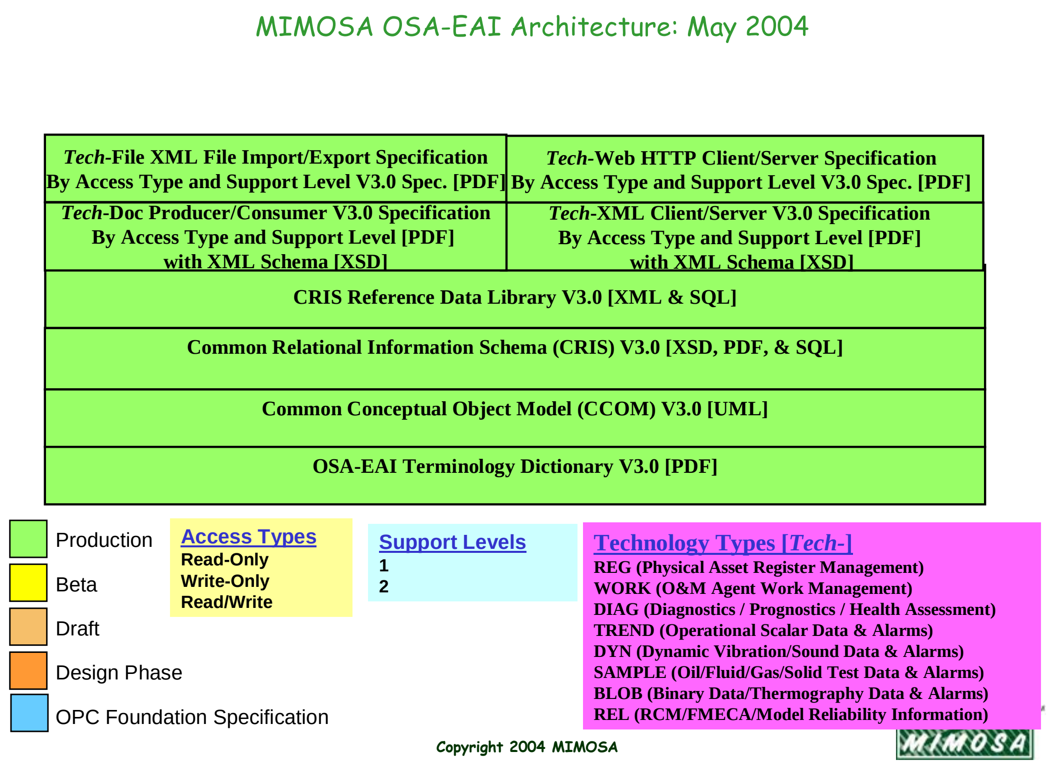

Figure 1. MIMOSA OSA-EAI Architecture Diagram

OSA-EAI Terminology Dictionary

To ease understanding among all parties using MIMOSA's OSA-EAI specification, MIMOSA provides a standard set of terminology in the OSA-EAI Terminology Dictionary. This provides the basic semantic descriptions used throughout the OSA-EAI specifications.

Common Conceptual Object Model (CCOM)

The Common Conceptual Object Model (CCOM) provides the basic conceptual model basis for OSA-EAI. Software designers familiar with class diagrams in Unified Modeling Language can utilize this model to understand the basic classes, major attributes, and relationships between classes in OSA-EAI. CCOM V3.0 is available in PDF document form and Visio (VSD) format.

Common Relational Information Schema (CRIS)

MIMOSA's Common Relational Information Schema (CRIS) provides a common implementation schema which allows information from many systems to be communicated and integrated. The schema is in a relational format, commonly used in most database systems. Each table in CRIS is assigned a unique table reference number. CRIS V3.0 is published in PDF document form, Word document form (DOC), and XML Schema (XSD) form.

Data sources which house asset information are not required to be physically re-designed to match CRIS, but must be able to translate their information into CRIS tables and columns. In order to ease this translation and for MIMOSA implementers who desire to implement a physical CRIS database, MIMOSA provides an ORACLE and Microsoft SQLServer table creation scripts.

CCOM/CRIS contain standard enterprise, site, functional segment, asset, and agent identification nomenclature. An enterprise is the corporate level of an organization, or the top organizational structure of a non-profit or military body. An enterprise is composed of many sites. The enterprise uniquely identifies the sites it manages. The enterprise ID is a globally-unique identifier and is defined as a 4-byte, non-negative integer assigned by MIMOSA. Normally, MIMOSA will issue one enterprise ID per corporation/organization. For some organizations, multiple enterprise IDs may be requested. In addition, MIMOSA will also assign the enterprise with a globally-unique, alpha-numeric user tag identifier value. This can be used in conjunction with the user tag identifier in the site table to form a globally unique text string. A representative from the registration authority for an organization should e-mail the MIMOSA Enterprise Registrar at **info@mimosa.org** with the name of the organization, requested enterprise User Tag Identifier, contact name, title, phone number, and e-mail address. The MIMOSA Enterprise Registrar will then assign the enterprise ID and enterprise User Tag Identifier and return the assigned non-negative integer and associated 8-byte string to the point of contact.

A “site” is what an enterprise defines as an entity which can be decomposed into segments and which generates new assets, agents, databases, and measurement locations. A given enterprise can contain many sites. A site can contain many segments. For facility applications, the “site” normally represents a building. For industrial applications, this entity normally represents a physical plant. For fleet applications, this entity normally represents a “mobile platform”, which could be an aircraft carrier or a tank. Each enterprise must uniquely identify its sites by a 4-byte, non-negative site ID integer to allow multiple sites to utilize the MIMOSA standards without fear of duplication when combining information at the enterprise level. The site ID is a 4-byte, non-negative integer. Each enterprise is free to assign site IDs in the range from 0 – 4,294967,295 (Hex "00000000" – "FFFFFFFF"). Once this assignment has been made, the number should not be changed.

Because CCOM and CRIS are designed for multi-site collaborative asset lifecycle management, nearly all table in CRIS (except MIMOSA-owned reference tables) include a reference to the enterprise/site/database. To keep the number of primary key columns to a minimum, CRIS V3 combines the enterprise ID (also 4-bytes) and the site ID into a fixed-length 16-character MIMOSA **site code**. This site code is composed of these two 4-byte non-negative integers which are converted into their hexadecimal format (resulting in 8 characters per integer) and then concatenated into a fixed-length 16-character string.

Version 3 of CCOM & CRIS have also added the ability to build Site templates. Templates are like "models" of sites which allows you to predefine characteristics of a Site, including its Segments, which software can utilize when instantiating a new Site which matches that template. In addition, they provides for a method of standard measurement location identification across various condition monitoring technologies. Trendable, scalar data such as operational temperatures, pressures, and loads are modeled in CCOM/CRIS. CCOM/CRIS support dynamic data, such as time waveforms and Fast Fourier Transforms (FFTs), which are used in vibration analysis and sound monitoring.   Binary data, known as Binary Large OBjects or (BLOBs), are supported for communicating drawings, reports, diagrams, and photographs. CCOM/CRIS also manages sampling test data results, such as used oil analysis test data and air quality monitoring data. CCOM/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, proposed), health and estimated asset life assessment, and recommendations. CCOM/CRIS model 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. CCOM/CRIS also provides the information framework for storing reliability data for assets.

CRIS Reference Database Specification

CCOM/CRIS contain many "type" classes/tables, which allow users to associate types of enterprises, sites, segments, assets, agents, measurement locations, engineering units, etc. with standard numeric codes, common throughout all their various systems.   MIMOSA generates and maintains industry-standard taxonomies and codes for most of these tables, but CCOM/CRIS allow both suppliers and end-users to add industry-specific and customer-specific entries to these tables. MIMOSA experts have generated a large reference database, the CRIS Reference Database Specification in XML, ORACLE, and SQLServer format. Version 3 of this database specification contains many useful codes which allow standardization across many disparate systems—even those from various countries. For example, the asset type table allows standard querying of common asset types such as “AC induction motor” which have unchanging three-integer unique identifiers. Other standard code tables include service segment, measurement location, engineering units, sampling test codes, diagnostic/prognostic event codes, health codes, failure codes, and root cause codes. This allows systems to search across various systems for common failures on certain equipment types.

***Tech*-XML Client/Server Interface Schemas**

A key component of MIMOSA’s Open System Architecture for Enterprise Application Integration (OSA-EAI) is the *Tech*-XML Client/Server interface schemas. These XML schemas provide a common set XML-based client/server interface definitions for various communication protocols. The *Tech*-XML interfaces specify the contents of a client/server data exchange, but do not specify the physical transport method.

***Tech*-Doc Producer/Consumer Interface Schemas**

For XML document-based systems, MIMOSA provides the *Tech*-Doc Producer/Consumer interface schemas, which publish/consume CRIS data from a given technology in an XML document format. *Tech*-Doc interfaces specify the contents of an XML document, but do not specify the physical storage and transport method of the produced/consumed document. *Tech*-Doc Consumer Read-only applications will utilize a *Tech*-Doc XML document, but not change its permanent data storage based on this information. *Tech*-Doc Consumer Write-only applications change their permanent data storage after successful file import.

***Tech*-XML and *Tech*-Doc Technology Types**

Both of these OSA-EAI interface technologies (*Tech*-XML & *Tech*-Doc) are further sub-divided into eight (8) Technology Types (see Figure 1 above). This allows developers to focus on supporting only what is relevant to their particular application area, such as vibration analysis or maintenance management. The technology types are listed in Table 1 below:

Technology Types	**Description**
Asset Register Management **(REG)**	Allows retrieval of "as-designed" segment hierarchical breakdown of facility, process, and machine systems, along with the "as-installed" asset information. Also allows access to name plate and image data on individual assets and models, including component part breakdowns. Used by: - OEM Model Information Systems - Asset Registry Information Systems - Maintenance Management Systems - Piping & Instrumentation Design Systems

Reliability **(REL)**	Allows retrieval of reliability study information related to a service segment, a model of an asset, or a serialized asset. Allows the retrieval of information associated with hypothetical causal networks, reliability-centered maintenance information, as well as actual failure events. Used by: - Reliability Information Systems - FMECA Analysis Systems - RCM Analysis Systems
Work Management **(WORK)**	Allows the creation and audit tracking of a new work request in a work management system for a service segment or a serialized asset. Allows the retrieval of work orders and work order steps, and actual work completed information.. Also allows the retrieval of pre-planned work packages ("solution packages"). Used by: - Maintenance Management Systems
Diagnostics / Prognostics / Health Assessment **(DIAG)**	Enables retrieval of human or "smart-agent" generated current and/or future proposed asset health states, current and/or future proposed diagnostic failure modes and casual trees, remaining useful life predictions, and recommendations. Also allows access to measurement evidence supporting the diagnoses/prognoses. Used by: - Diagnostic Systems - Prognostic Systems
Dynamic Vibration/Sound Condition Monitoring **(DYN)**	Enables the creation and retrieval of historical dynamic measurements (used with vibration and sound monitoring and including frequency spectra measurements and time waveforms), abnormal data alarms, and operational event logs. Used by: - Vibration Condition Monitoring Systems - Sound Condition Monitoring Systems
Static "Trendable" Condition Monitoring **(TREND)**	Enables the creation and retrieval of historical scalar measurements, abnormal data alarms, and operational event logs. Used by: - Process Data Historians - Process Condition Monitoring Systems - Used by Operational Data Systems
Oil/Fluid/Gas/ Solid Tests Condition Monitoring **(SAMPLE)**	Enables the creation and retrieval of historical fluid, air, and solid sampling data, abnormal data alarms, and operational event logs. Used by: - Oil Sampling Condition Monitoring Systems - Air Sampling Condition Monitoring Systems - Solid Sampling Condition Monitoring Systems

Binary Data / Thermography Condition Monitoring **(BLOB)**	Enables the creation and retrieval of historical binary large objects (BLOB) measurements (used with thermography and imaging monitoring), abnormal data alarms, and operational event logs Used by: - Thermographic Condition Monitoring Systems - Image Monitoring Systems

Table 1. OSA-EAI Application Technology Packages

To refer to the entire set of 8 Application Technology Packages, the italicized prefix "*Tech-"* is used. Each vertical application technology only has certain CRIS tables which are relevant. This allows implementers to pair down the entire CRIS specification to only the application-specific tables. To support this, MIMOSA publishes the ***Tech*-XML Support Level *n* CRIS V3.0 Subset Specification** for *Tech*-XML developers and the ***Tech*-Doc Support Level *n* CRIS V3.0 Valid Table Entity Specification** for *Tech*-Doc developers.

Tech-Web Specifications

The OSA-EAI *Tech*-Web specifications are used for building a server or a client which can communicate data and information between condition monitoring systems, diagnostic systems, reliability management systems, registry management systems, and work management systems via HTTP or HTTPS (secure) protocol using XML messages. The interfaces are defined using XML Schema and specified in a client/server fashion.

The communications between the client and the server (XML Dialogs) will use industry standard Web services technologies summarized in Table 3 below. These web services are hardware, operating system, and programming language independent and are design to operate on nearly all data networks.

Function	Technology	Description
Description	XML Schema	Structured metadata about the interface
Encoding	XML	Standard, platform-neutral way to encode data exchanged between client and server
Transport	HTTP	Standard, stateless transport protocol

Table 3. Demonstration Web Services Utilization Chart

Tech-Web servers must comply with MIMOSA's **Tech-Web Client & Server HTTP Binding Specification** supporting interfaces defined for a given technology type (TREND, REG, WORK, etc), access type (Read-Only, Write-Only, or Read/Write), support level (level 1 or 2), and version level (3.0).

***Tech*-File Specifications**

The *Tech*-File specifications are used for building an XML document export or import applications which can communicate data and information between condition monitoring systems, diagnostic systems, reliability management systems, registry management systems, and work management systems. The interfaces are defined using XML Schema and specified in a XML file import/export fashion.

Tech-File export applications must comply with MIMOSA's **Tech-File Export Specification** supporting XML file exports defined for a given technology type (TREND, REG, WORK, etc), access type (Read-Only, Write-Only, or Read/Write), support level (level 1 or 2), and version level (3.0). *Tech*-File import Read-only applications will import a standard XML export file, but not change its permanent data storage based on this information. *Tech*-File import Write-only applications change their permanent data storage after successful file import.

OSA-EAI Interface Catalog and OSA-EAI Compliance Process

All OSA-EAI interfaces are assigned a 9-digit number, published in the **OSA-EAI V*n.n* Interface Catalog**. Applications may support one or more OSA-EAI interfaces. System designers need to decide which interfaces are applicable for a given system and then write these OSA-EAI interfaces. To ensure acceptance by the user community at large, the use of the MIMOSA logo and the trademark MIMOSA Compliant® must be limited to those products that have met MIMOSA's requirements for compliance. Product buyers must have confidence that the logo and trademark on a product means that the product has the “stamp of approval” from MIMOSA. MIMOSA has chosen to support compliance and certification through a process of self-testing using toolkits and/or a certification procedure supplied by MIMOSA. Both servers and clients can be certified. Since the needs of client applications vary greatly and no other application is dependent on them, there is no testing tool provided for testing client applications.

MIMOSA sponsors have access to compliant self-certification tools and may submit each major release of their products to MIMOSA as MIMOSA-compliant to a specific set of OSA-EAI interfaces. After MIMOSA reviews the submission and validates the compliance, sponsors are granted permission to mark their products with the MIMOSA-owned registered trademark "MIMOSA-compliant®" brand. Companies which are not MIMOSA sponsors or companies who produce products which have not complied with MIMOSA's compliancy requirements are not granted permission to use "MIMOSA-compliant" on their products. MIMOSA will post all currently certified *MIMOSA Compliant*® products on its website, listing the set of OSA-EAI interfaces supported.