Open System Architecture For Enterprise Application Integration

# Technical Architecture Summary

## 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 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.

# 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 OpenMRO™ -- 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.

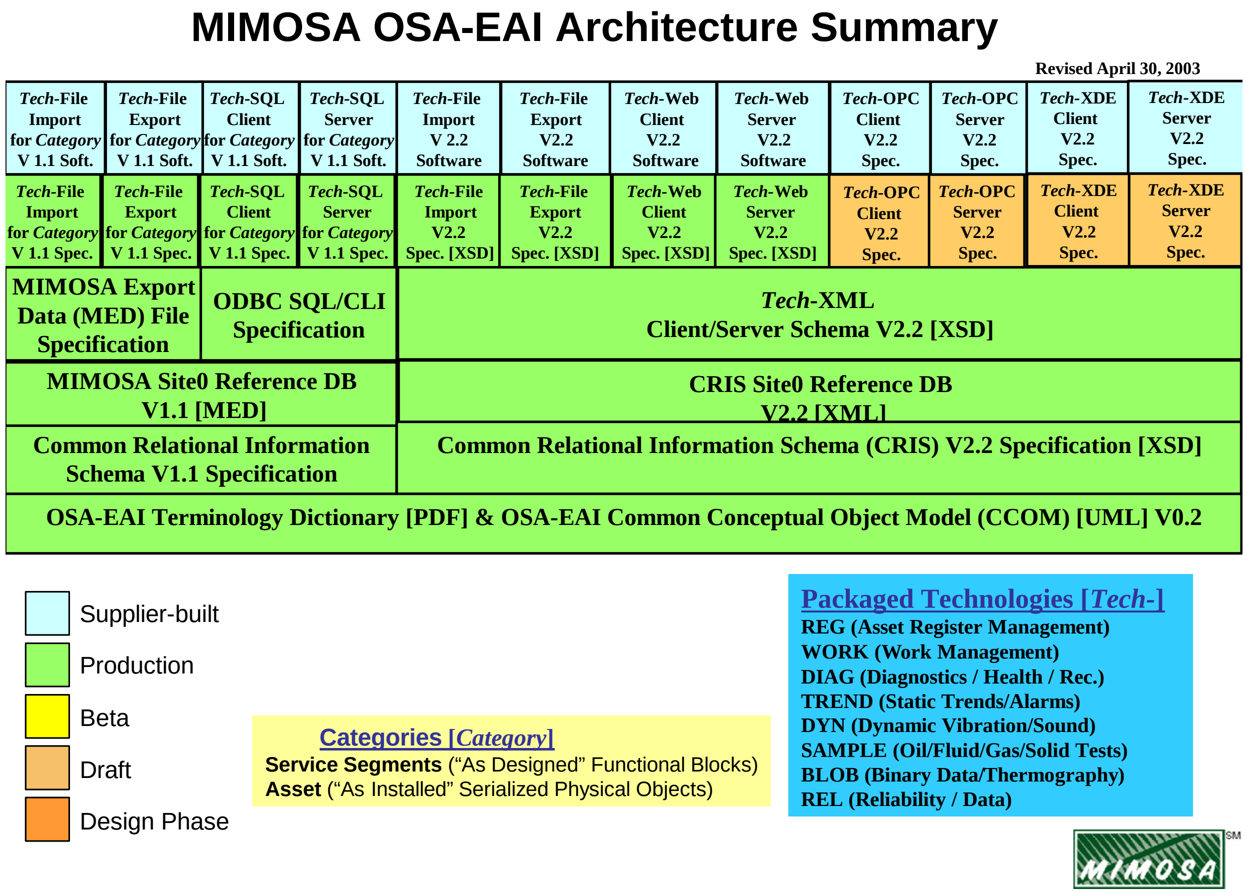


**Figure 1. MIMOSA Architecture Diagram**

To ease understanding among all parties using MIMOSA's specifications, 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's OSA-EAI specification is built upon a common information schema which allows 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 temperatures, pressures, and loads are modeled in CRIS. CRIS supports 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. CRIS also manages sampling test data results, such as used oil analysis test data and air quality monitoring data. 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. 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 also provides the information framework for storing reliability data for assets. 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 specification contains many useful codes which allow standardization across many disparate systems—even those from various countries. For example, the Site Zero database contains a standard universal asset type taxonomy, which allows standard querying of common asset types such as "AC induction motor" which have never-changing 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.

Data sources which house asset information are not required to be physically re-designed to match CRIS, but must be able to translated into CRIS tables and columns. In order to ease this translation and for MIMOSA implementers who desire to implement a physical CRIS database, MIMOSA also provides populated CRIS V2.2 reference database to its members in the following popular database formats: Microsoft Access, ORACLE, and Microsoft SQLServer.

A key component of MIMOSA's Open System Architecture for Enterprise Application Integration (OSA-EAI) is the *Tech*-XML Client/Server schema. This XML schema provides a common set XML-based client/server interface definitions for various protocols.

Built on the foundation of these specifications, MIMOSA has defined six types of interfaces which can be utilized for OSA-EAI integration:

? *Tech*-File Import – for importing CRIS data in a file format

? *Tech*-File Export – for exporting CRIS data in a file format

? *Tech*-SQL Client – for reading CRIS data from an ODBC data server

? *Tech*-SQL Server – for publishing CRIS data using an ODBC data server

? *Tech*-Web Client – for querying/creating CRIS data from an XML Web data server

? *Tech*-Web Server – for publishing/creating CRIS data using an XML Web data server

Each of these OSA-EAI interface types are further sub-divided into eight (8) Application Technology Packages. This allows developers to focus on supporting only what is relevant to their particular application area, such as vibration analysis or maintenance management. The packages are listed in Table 1 below:

| **Application Technology Packages** | **Description** |
|---|---|
| Asset Register Management<br><br>**(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.<br>Used by:<br>- OEM Model Information Systems<br>- Asset Registry Information Systems<br>- Maintenance Management Systems<br>- Piping & Instrumentation Design Systems |
| Work Management<br><br>**(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").<br>Used by:<br>- Maintenance Management Systems |
| Diagnostics / Health / Rec.<br><br>**(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.<br>Used by:<br>- Diagnostic Systems<br>- Prognostic Systems |
| Dynamic Vibration/Sound<br><br>**(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.<br>Used by:<br>- Vibration Condition Monitoring Systems<br>- Sound Condition Monitoring Systems |
| Static Trends/Alarms<br><br>**(TREND)** | Enables the creation and retrieval of historical scalar measurements, abnormal data alarms, and operational event logs.<br>Used by:<br>- Process Data Historians<br>- Process Condition Monitoring Systems<br>- Used by Operational Data Systems |
| Oil/Fluid/Gas/Solid Tests<br><br>**(SAMPLE)** | Enables the creation and retrieval of historical fluid, air, and solid sampling data, abnormal data alarms, and operational event logs.<br>Used by:<br>- Oil Sampling Condition Monitoring Systems<br>- Air Sampling Condition Monitoring Systems<br>- Solid Sampling Condition Monitoring Systems |
| Binary Data/Thermography | 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 |

| | |
|---|---|
| **(BLOB)** | Used by:<br>- Thermographic Condition Monitoring Systems<br>- 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 OSA-EAI Application Technology CRIS V2.2 Cross-Reference Matrix in PDF format.

Each OSA-EAI interface specification is given a unique number as shown in Table 2 below.

| | | | **Interface Type** | | | |
|---|---|---|---|---|---|---|
| **Application Technology Package** | ***Tech-* File Import** | ***Tech-* File Export** | ***Tech-* SQL Client** | ***Tech-* SQL Server** | ***Tech-* Web Client** | ***Tech-* Web Server** |
| ? **TREND** (static value, alarms) | **#1110111** (reads MED-formatted, segment-centric, trend data-related files from CRIS V1.1)<br><br>**#2110111** (reads MED-formatted, asset-centric, trend data-related files from CRIS V1.1)<br><br>* **#9110122** (reads a XML-formatted, segment and/or asset-centric, trend data-related file from CRIS V2.2) | **#1120111** (writes MED-formatted, segment-centric, trend data-related files from CRIS V1.1)<br><br>**#2120111** (writes MED-formatted asset-centric, trend data-related files from CRIS V1.1)<br><br>* **#9120122** (writes a XML-formatted, segment and/or asset-centric, trend data-related file from CRIS V2.2) | **#1310111** (issues ODBC queries for segment-centric, trend-related data from CRIS V1.1)<br><br>**#2310111** (issues ODBC queries for asset-centric, trend-related data from CRIS V1.1) | **#1320111** (ODBC-server for segment-centric queries of trend-related data from CRIS V1.1)<br><br>**#2320111** (ODBC-server for asset-centric queries of trend-related data from CRIS V1.1) | * **#9610122** (issues HTTP XML requests for segment and/or asset-centric trend-related data from CRIS V2.2) | * **#9620122** (acknowledges HTTP XML requests for segment and/or asset-centric trend-related data from CRIS V2.2) |

| | | | Interface Type | | | |
|---|---|---|---|---|---|---|
| **Application Technology Package** | ***Tech-File Import*** | ***Tech-File Export*** | ***Tech-SQL Client*** | ***Tech-SQL Server*** | ***Tech-Web Client*** | ***Tech-Web Server*** |
| ? **DYN** (dynamic vibration, sound, electric current) | **#1110211** (reads MED-formatted, segment-centric, dynamic data-related files from CRIS V1.1)<br>**#2110211** (reads MED-formatted, asset-centric, dynamic data-related files from CRIS V1.1)<br>* **#9110222** (reads a XML-formatted, segment and/or asset-centric, dynamic data-related file from CRIS V2.2) | **#1120211** (writes MED-formatted, segment-centric, dynamic data-related files from CRIS V1.1)<br>**#2120211** (writes MED-formatted asset-centric, dynamic data-related files from CRIS V1.1)<br>* **#9120222** (writes a XML-formatted, segment and/or asset-centric, trend data-related file from CRIS V2.2) | **#1310211** (issues ODBC queries for segment-centric, dynamic-related data from CRIS V1.1)<br>**#2310211** (issues ODBC queries for asset-centric, dynamic-related data from CRIS V1.1) | **#1320211** (ODBC-server for segment-centric queries of dynamic-related data from CRIS V1.1)<br>**#2320211** (ODBC-server for asset-centric queries of dynamic-related data from CRIS V1.1) | * **#9610222** (issues HTTP XML requests for segment and/or asset-centric dynamic-related data from CRIS V2.2) | * **#9620222** (acknowledges HTTP XML requests for segment and/or asset-centric dynamic-related data from CRIS V2.2) |
| ? **SAMPLE** (oil, fluid, gas tests) | **#9110322** (reads a XML-formatted, segment and/or asset-centric, sample data-related file from CRIS V2.2) | **#9120322** (writes a XML-formatted, segment and/or asset-centric, sample data-related file from CRIS V2.2) | | | **#9610322** (issues HTTP XML requests for segment and/or asset-centric sample-related data from CRIS V2.2) | **#9620322** (acknowledges HTTP XML requests for segment and/or asset-centric sample-related data from CRIS V2.2) |
| ? **BLOB** (Binary large objects/ thermo-graphy) | **#9110422** (reads a XML-formatted, segment and/or asset-centric, BLOB data-related file from CRIS V2.2) | **#9120422** (writes a XML-formatted, segment and/or asset-centric, BLOB data-related file from CRIS V2.2) | | | **#9610422** (issues HTTP XML requests for segment and/or asset-centric BLOB-related data from CRIS V2.2) | **#9620422** (acknowledges HTTP XML requests for segment and/or asset-centric BLOB-related data from CRIS V2.2) |
| ? **REG** | * **#9110522** | * **#9120522** | | | * **#9610522** | * **#9620522** |

| | | | Interface Type | | | |
|---|---|---|---|---|---|---|
| **Application Technology Package** | ***Tech- File Import*** | ***Tech- File Export*** | ***Tech- SQL Client*** | ***Tech- SQL Server*** | ***Tech- Web Client*** | ***Tech- Web Server*** |
| (asset manageme nt registry) | (reads a XML-formatted, segment and/or asset-centric, registry data-related file from CRIS V2.2) | (writes a XML-formatted, segment and/or asset-centric, registry data-related file from CRIS V2.2) | | | (issues HTTP XML requests for segment and/or asset-centric registry-related data from CRIS V2.2) | (acknowledges HTTP XML requests for segment and/or asset-centric registry-related data from CRIS V2.2) |
| ? **DIAG** (diag-nostics, recomm.) | **#9110622** (reads a XML-formatted, segment and/or asset-centric, diagnostic data-related file from CRIS V2.2) | **#9120622** (writes a XML-formatted, segment and/or asset-centric, diagnostic data-related file from CRIS V2.2) | | | *** #9610622** (issues HTTP XML requests for segment and/or asset-centric diagnostic-related data from CRIS V2.2) | *** #9620622** (acknowledges HTTP XML requests for segment and/or asset-centric diagnostic-related data from CRIS V2.2) |
| ? **REL** (reliability data, failure modes) | **#9110722** (reads a XML-formatted, segment and/or asset-centric, reliability data-related file from CRIS V2.2) | **#9120722** (writes a XML-formatted, segment and/or asset-centric, reliability data-related file from CRIS V2.2) | | | **#9610722** (issues HTTP XML requests for segment and/or asset-centric reliability-related data from CRIS V2.2) | **#9620722** (acknowledges HTTP XML requests for segment and/or asset-centric reliability-related data from CRIS V2.2) |
| ? **WORK** (work manage-ment) | **#9110822** (reads a XML-formatted, segment and/or asset-centric, work data-related file from CRIS V2.2) | **#9120822** (writes a XML-formatted, segment and/or asset-centric, work data-related file from CRIS V2.2) | | | *** #9610822** (issues HTTP XML requests for segment and/or asset-centric work-related data and actions from CRIS V2.2) | *** #9620822** (acknowledges HTTP XML requests for segment and/or asset-centric work-related data and actions from CRIS V2.2) |

**Table 2. MIMOSA OSA-EAI Interface Catalog**

## 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><b><Font size=”+1”>Brief Background on XML</font></b></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 Tech-Web Specifications

MIMOSA's Tech-Web specifications are used for communicating data and information between condition monitoring systems, diagnostic systems, asset 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 | Required for CONOPS? |
|---|---|---|---|
| Description | XML Schema | Structured metadata about the interface | Yes |
| Encoding | XML | Standard, platform-neutral way to encode data exchanged between client and server | Yes |
| Transport | HTTP | Standard, stateless transport protocol | Yes |

**Table 3. Demonstration Web Services Utilization Chart**