MIMOSA

Getting started guide

January 29, 2021

This document outlines the various MIMOSA and OpenO&M initiatives and how they work together and their role in the OIIE. It provides the basic information for understanding the purpose of each component and how they fit together with references to the detailed specifications.

Contents

© 1998 - 2021 MIMOSA. All rights reserved.

Glossary

BOD
Business Object Document

CCOM
Common Conceptual Object Model

CIR
Common Interoperability Register

ISBM
Information Service Bus Model

ISDD
Industry Standard Datasheet Definition

SDAIR
Structured Digital Asset Interoperability Register

OGI
Oil & Gas Interoperability

OIIE
Open Industrial Interoperability Ecosystem

© 1998 - 2021 MIMOSA. All rights reserved.

Overview

MIMOSA™ is a not-for-profit industry trade association, incorporated in California, United States of America, dedicated to developing and encouraging the adoption of open, supplier-neutral IT and IM standards enabling digitalization and interoperability for asset life-cycle management. MIMOSA standards support key requirements for critical infrastructure management. For example, MIMOSA standards and specifications enable a digital twin to be defined and maintained on a supplier-neutral basis, while also using that digital twin to provide context for big data (IIOT and other sensor-related data) and analytics. The sensor-based flows and analytics include industry leading standards for Condition Based Maintenance (CBM).

MIMOSA manages the development, validation, and maintenance of a specification for a supplier neutral industrial digital ecosystem, the Open Industrial Interoperability Ecosystem™ (OIIE™) in cooperation with other industry associations. The OIIE is defined through industry standard OIIE Use Cases and a standard portfolio of supplier neutral industry standards. These use cases are specified based on the standard OIIE Use Case Architecture, which captures requirements ranging from a basic user story to the details needed for actual code implementation. OIIE Use Cases are sponsored by individual members or groups of members, thereby spreading the risk and the cost of joint industry projects.

MIMOSA manages the associated OIIE piloting environment using commercial cloud infrastructure, in cooperation with other industry associations, including the industry standards development organizations in the OpenO&M Initiative™. The OIIE Oil and Gas Interoperability (OGI) Pilot is an instance of the OIIE which includes processes, systems and component specifications which are used in both the oil and gas sector and other asset intensive industries. Applications and systems from multiple suppliers are validated to support the OIIE use cases, collaboratively improving the applications, systems, use cases and the associated technical specifications.

ISO TC 184/WG 6 uses the OIIE OGI Pilot in the development of ISO 18101, the asset intensive industry interoperability standard, which is based on the OIIE. The technical reports generated by the OIIE OGI Pilot team are turned into formal ISO Technical Reports which are then used to shape the subsequent parts of ISO 18101 if that is deemed appropriate. The MIMOSA solutions process seeks to avoid “re-inventing wheels” by leveraging existing standards such as ISA-95, ISO 8000, ISO 15926, and ISO 18435. MIMOSA was an early adopter of XML and is now adding JSON data formats for sensor-based data. The focus is on solving business problems. An example of an open MIMOSA standard is CCOM – the Common Conceptual Object Model. This standard serves as a data model for the exchange of asset life-cycle information, facilitating standards-based interoperability between systems. MIMOSA also maintains and publishes the Open Systems Architecture for Condition Based Maintenance (OSA-CBM), the main input for ISO 13374.

Standards, Specifications and Activities

The following is a list of primary core standards, specifications, and activities that MIMOSA and OpenO&M initiative have produced or are work in progress.

- *OIIE Information and Systems Architecture*
- *OIIE Use Case Architecture*
- *OIIE OGI Pilot*
- *MIMOSA CCOM*
- *MIMOSA SDAIR*
- *MIMOSA Services Directory*
- *MIMOSA ISDD Project*

© 1998 - 2021 MIMOSA. All rights reserved.

- *OpenO&M ISBM*
- *OpenO&M ws-CIR*

The following briefly describes these different standards, specifications and activities and their role in the OIIE.

OIIE Information and Systems Architecture

Taken as a whole, the data required for asset lifecycle management is complex, incorporating asset information from a variety of information systems including OEMs, logistics, condition assessment, operations, and maintenance information systems. Moreover, the information is typically further separated by type: manufacturer's nameplate data, as-engineered, as-designed, as-built, as-maintained, operational, condition monitoring, and asset diagnostic/health and reliability data are all stored and accessed separately through different systems. This has caused problems for operators, maintenance personnel, logistic managers, OEM's, parts suppliers, and engineers who want to have information about an asset and its condition at their fingertips when they need it. In this state it is difficult, if not impossible, to obtain a wholistic view of the information suitable for making intelligent asset management decisions. Even when the systems can be accessed through the same interface, it usually requires customized and time-consuming efforts to integrate disparate databases which use proprietary programs with different data models.

The OIIE is a specification for a supplier neutral industrial digital ecosystem which leverages standard specifications for APIs and transport protocols, including the OpenO&M ISBM, which is linked to the IEC 62264 MSM and ISA 95 standards. The OIIE also leverages Industry Standard Datasheet Definitions (ISDDs), which are based on published Industry Standard Datasheets (ISDs) published by authoritative groups such as ISA, PDEX, API and PIP. MIMOSA updates and maintains the ISBM specification in cooperation with members of the OpenO&M Initiative including ISA, MESA/B2MML, OAGi and OPC Foundation. MIMOSA develops and publishes ISDDs in cooperation with the organizations maintaining and publishing the ISDs. The OIIE and the OGI Pilot takes a Use Case-based approach built around the OIIE Use Case Architecture, with prioritized use cases identified by industry partners. OIIE Compliant applications and systems must be able to publish and consume the standard OIIE Scenarios and Events using the ISBM and ISDDs and support of the OIIE Use Cases for which they wish to claim compliance.

The high-level systems architecture of OIIE detailing generic system classes as well the connectivity and services within the system of systems is shown in Figure 1. An information service bus approach is used for connectivity between applications. The goal of the information service bus is to support data exchanges in use cases that are designed around an event-driven architecture or service-oriented architecture within a single, consistent model. Recognition is given to the fact that automation and control networks for Layer 2 operations and below are typically segregated from corporate networks by a DMZ. A shared reference data management platform provides classes, taxonomies and ontologies for shared data context in data exchanges.

For more details on OIIE systems architecture, please visit *here*.

© 1998 - 2021 MIMOSA. All rights reserved.

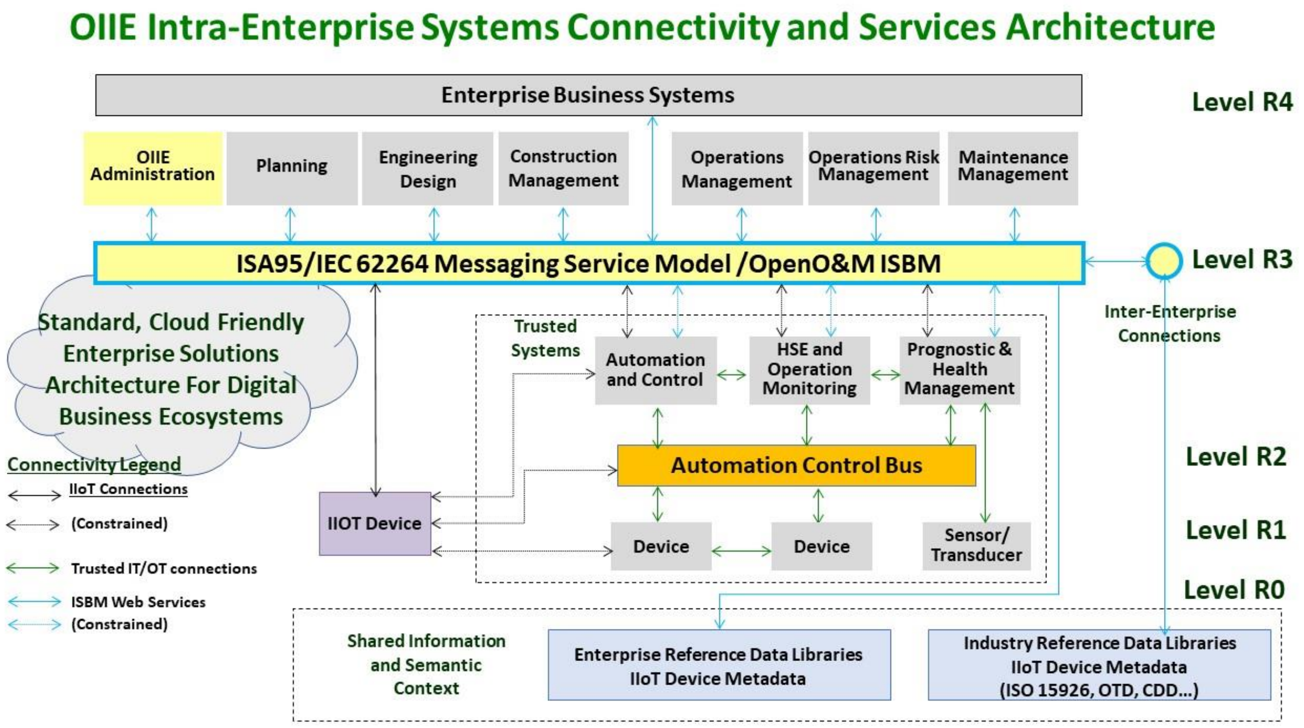


Figure 1 OIIE Information and Systems Architecture

OIIE Use Case Architecture

The Open Industrial Interoperability Ecosystem (OIIE) is built around a set of interoperability use cases that describe how industrial systems are to interact to achieve functionality desired/required by organizations involved in asset lifecycle management. The use cases allow organizations (EPCs, O/Os, Software Vendors, etc.) to declare conformance to specific reusable chunks of functionality, with which its systems can interoperate. The set of use cases is incrementally extended to incorporate new functionality as each set is validated by pilots, such as the OGI Pilot, with the inclusion of new use cases guided by industry partners.

Each use case conforms to the OIIE Use Case Architecture, which defines a standardized breakdown of Use Cases into smaller reusable parts, as well as a top-level overview of a Use Case or group of connected Use Cases. This breakdown forms a 3+1 level architecture, totaling 4 main components: Use Cases, Scenarios, Events, and User Stories. Each of the first two components decompose into the next, i.e., Use Cases decompose into Scenarios and Scenarios decompose into Events, while the fourth, User Stories, forms the “+1” as they can cross the other layers to illustrate specific events or whole use cases as required to achieve their purpose. The

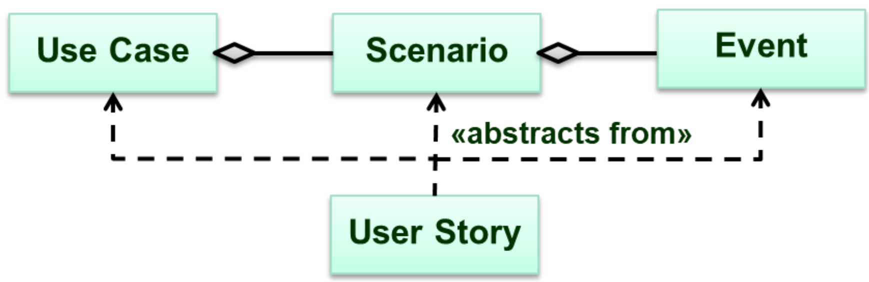


Figure 2 Relationships between Use Case Architecture Components

components of the Use Case Architecture and the relationships between them are summarized in Figure 2.

© 1998 - 2021 MIMOSA. All rights reserved.

Use Cases describe common interactions and context to achieve an interoperability goal and are decomposed into Scenarios. Each Scenario provides additional details and requirements on how to achieve an interaction based on a specific group of Events. The Event descriptions detail specific message exchanges and their requirements but are general enough to support different realizations of the exchanges over different protocols and data formats. Finally, these three components are tied together by User Stories, which abstract from the underlying components to provide a higher-level overview of interactions and connect Use Cases in a logical flow. An overview of the Use Case Architecture is shown in Figure 3.

Beginning in 2007, representatives from the Oil & Gas and Petrochemical industries participated in an *OpenO&M* End-User Advisory Group to provide the highest valued Use Cases with interoperability scenarios required for organizations to meet their business objectives. These have been continuously developed and refined through the *OGI Pilot*.

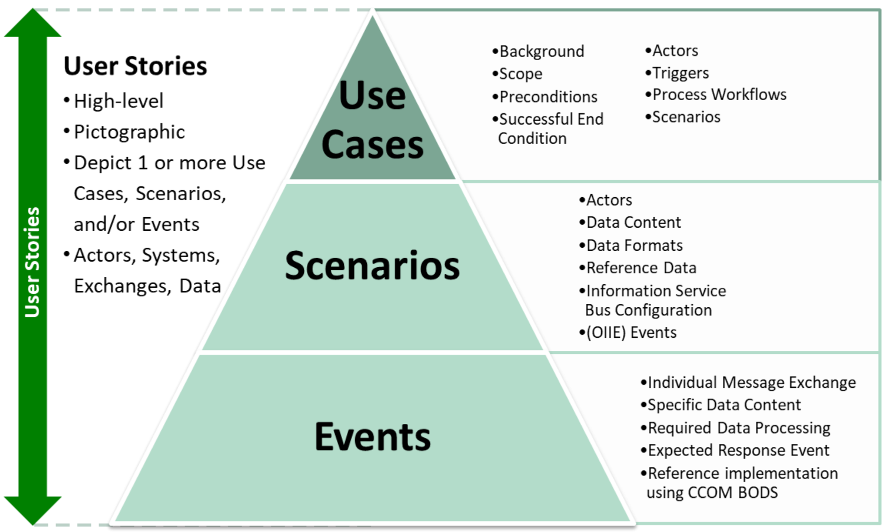


Figure 3 OIIE Use Case Architecture Overview

Current Use Cases

- OIIE Use Case 1 – Information Handover from EPC to O/O
- OIIE Use Case 2 – Engineering Updates
- OIIE Use Case 3 – Field Changes to Plant/Facility Engineering
- OIIE Use Case 4 – Online Product Data Library Management
- OIIE Use Case 5 – Asset Installation/Removal Updates
- OIIE Use Case 6 – Preventive Maintenance Triggering
- OIIE Use Case 7 – Condition-Based Maintenance Triggering
- OIIE Use Case 8 – Early Warning Notifications
- OIIE Use Case 9 – Incident Management/Accountability
- OIIE Use Case 10 – Information Provisioning of O&M Systems
- OIIE Use Case 11 – Enterprise Reference Data Library Management

© 1998 - 2021 MIMOSA. All rights reserved.

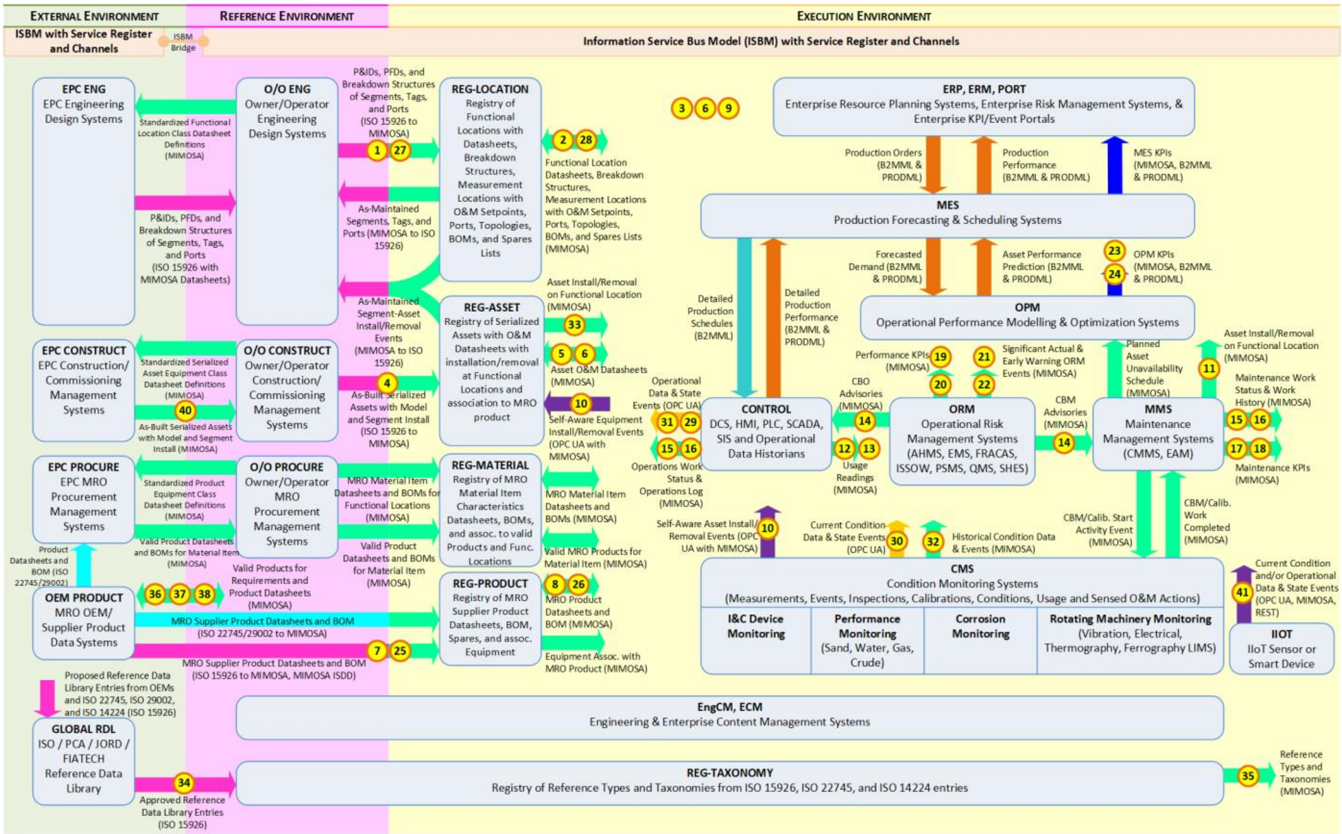
- OIIE Use Case 12 – RFI and RFI Response for Models Meeting Requirements (Greenfield and Brownfield)
- OIIE Use Case 13 – Lockout-Tagout
- OIIE Use Case 14 – Condition-Based Maintenance Data Acquisition
- OIIE Use Case 15 – Capital Project Asset Installation

Pending Use Cases

These are use cases in which owner/operators have expressed interest, but detailed analysis work has not yet been completed.

- OIIE Use Case – Asset Configuration Management
- OIIE Use Case – Reference Data Standardization
- OIIE Use Case – Capital Project Tag Generation

Each of the above Use Cases have a number of interoperability scenarios that indicate the source and target systems, information payload and format, triggers and transfer mechanism. The OIIE Systems Landscape Data Flow Diagram below provides a single visual rendering of the scenarios with the relevant system blocks that represent categories of data/information repositories, and arrows that represent the standardized information data flow scenarios between the system blocks.



For more details on OIIE use case architecture and to explore the latest OIIE use case package, please visit *here*.

© 1998 - 2021 MIMOSA. All rights reserved.

OIIE OGI Pilot

In order to validate the OIIE Specification, which is a primary informative reference for the ISO 18101-TS, and the ability of included COTS applications from multiple industry suppliers to use the OIIE to properly support the included OIIE Use Cases, MIMOSA manages the associated, OIIE Piloting Environment using commercial cloud infrastructure. The Oil and Gas Interoperability (OGI) Pilot is an instance of the OIIE which runs in the OIIE Piloting Environment which includes processes, systems and component specifications which are used in both the Oil and Gas Industry and other asset intensive process-oriented industries. Applications and systems (both COTS and Open Source) from multiple suppliers are validated to support the included OIIE Use Cases, while also helping to improve the use cases themselves as well as the associated technical specifications.

The OGI Pilot was initiated in 2010 and it continues to be managed by MIMOSA, functioning as an R&D test-bed for what has now grown into the OIIE, based on Oil and Gas industry assets. Various owner/operators have also run internal Proof of Concept exercises and analysis in 2014 and 2015. The OIIE OGI Pilot is run in a programmatic fashion, split into a continuing series of project phases, each of which incrementally builds on the former phases. At the end of each phase, a Technical Report is generated which includes Lessons Learned and a Gap Analysis, which is used to develop recommendations for Next Steps to drive incremental updates of the OIIE, the included specifications and future pilot phases.

The OGI Pilot incorporates full Life-cycle Information Management along with System of Systems Interoperability, as it establishes the OGI Ecosystem, which is fully defined by published, supplier-neutral standards, specifications and conventions. The OGI Pilot is driven by prioritized industry "fully dressed use cases" (OGI Use Cases), which are defined all the way to the required, standardized services. The OGI Pilot provides a practical proving ground, where the basis for System of Systems Interoperability can be established and validated at scale, against specific industry defined data sets, which are created and managed by EPC firms and validated by Owner/Operators.

MIMOSA, its members, and OIIE OGI Pilot participants recently celebrated the success of Phase 3.1 of the OIIE OGI Pilot. Phase 3.1 expanded the scope of previous Pilot phases dramatically, covering OIIE Use Cases from a range of activities across CAPEX and OPEX as shown in the Figure 4. A video demonstration of the pilot is available on *YouTube* and *Vimeo*.

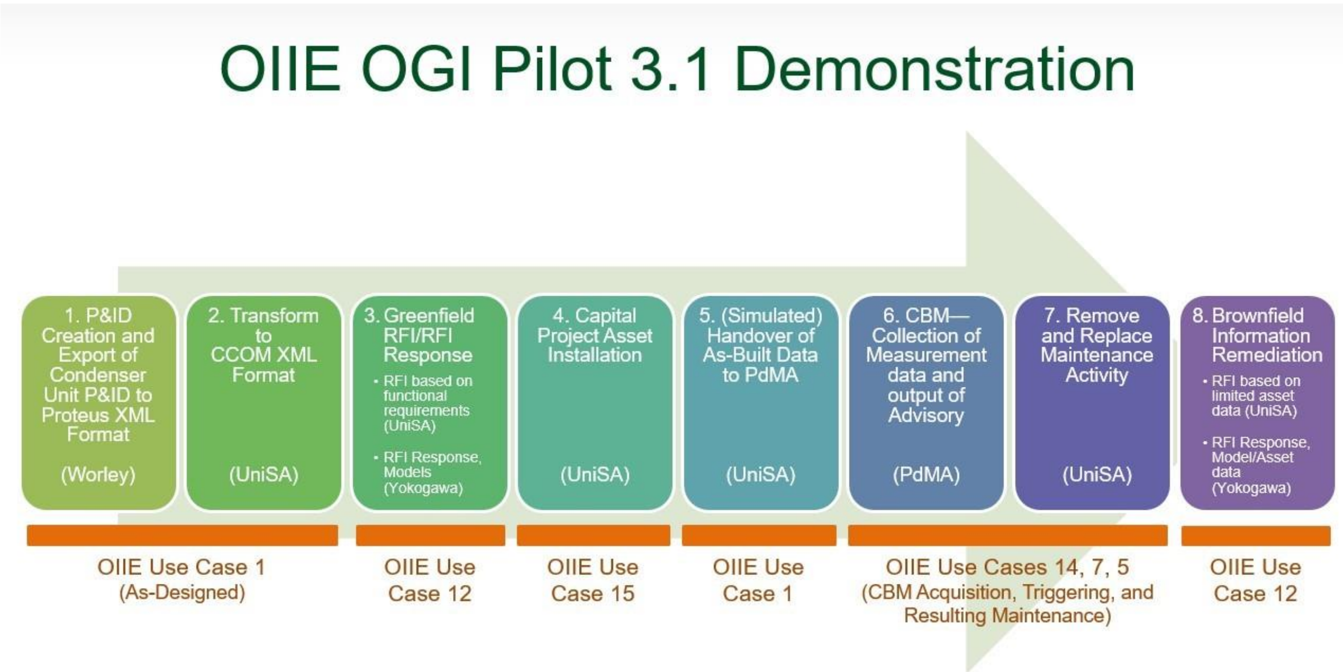


Figure 4 OGI Pilot 3.1 Demonstration Flow

© 1998 - 2021 MIMOSA. All rights reserved.

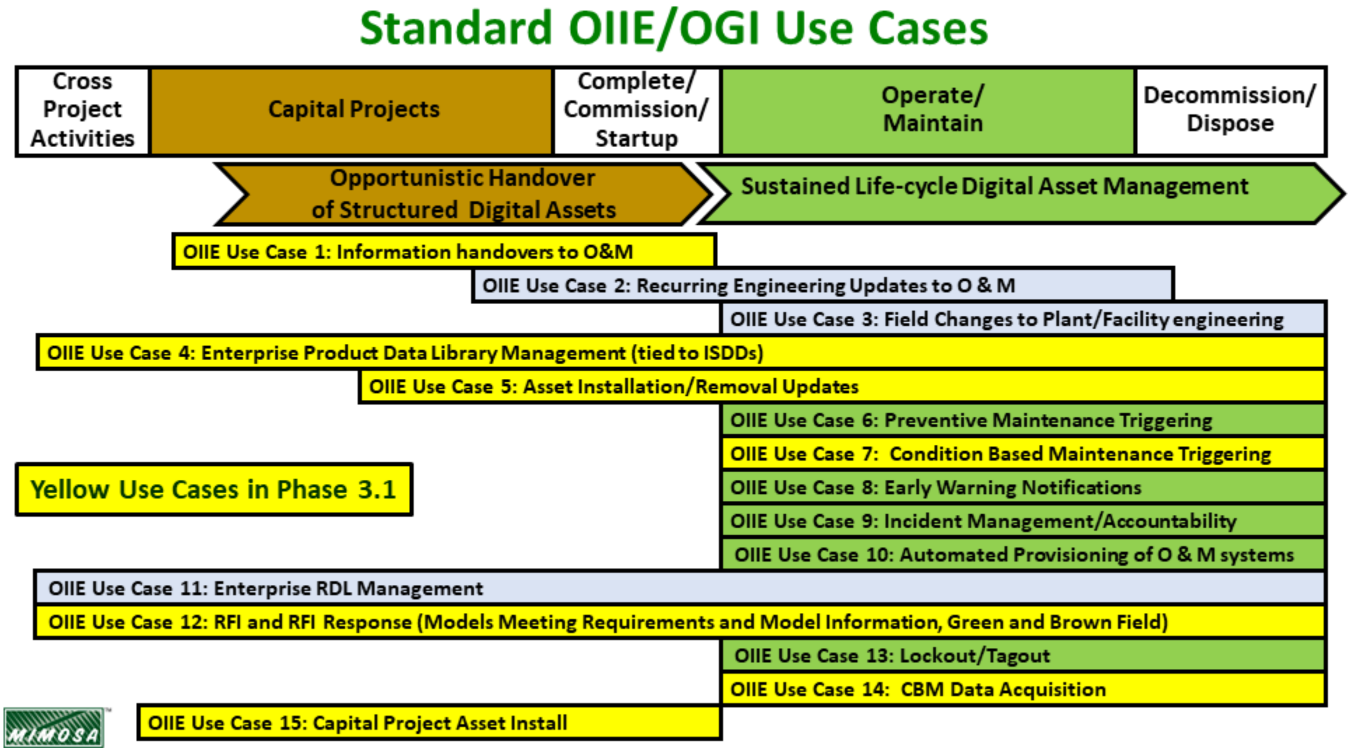


Figure 5 OIIE Use Cases demonstrated in OIIE OGI Pilot 3.1

The OIIE Use Cases highlighted in yellow in the Figure 5, below, were demonstrated in OGI Pilot Phase 3.1.

For more details on OGI Pilot, please visit *here*.

MIMOSA CCOM

MIMOSA CCOM (and its relational counterpart CRIS) was an integral element of MIMOSA's Open Systems Architecture for Enterprise Application Integration (OSA-EAI), which aimed to link the separate information islands of engineering, operations, and reliability information into a network that supported the common understanding and utilization of the information. MIMOSA CCOM now fills the same role within the OpenO&M Initiative's OIIE Specification. 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, by building bridges to the proprietary data stores through the OIIE, the specialised proprietary systems within an OIIE instance can exchange information to provide combined value that far exceeds the individual value of each system.

MIMOSA CCOM (Common Conceptual Object Model) serves as an information model for the exchange of asset lifecycle information, including engineering, asset, configuration, operation and condition, etc., required for the Operation and Maintenance of plants and complex facilities, but which can be used to provide the contextual basis for defining and maintaining Digital Twins and performing Big Data Analytics. Its core mission is to facilitate standards-based interoperability between systems: providing an XML model (among others) to allow systems to electronically exchange data through adaptors, as illustrated in Figure 6.

© 1998 - 2021 MIMOSA. All rights reserved.

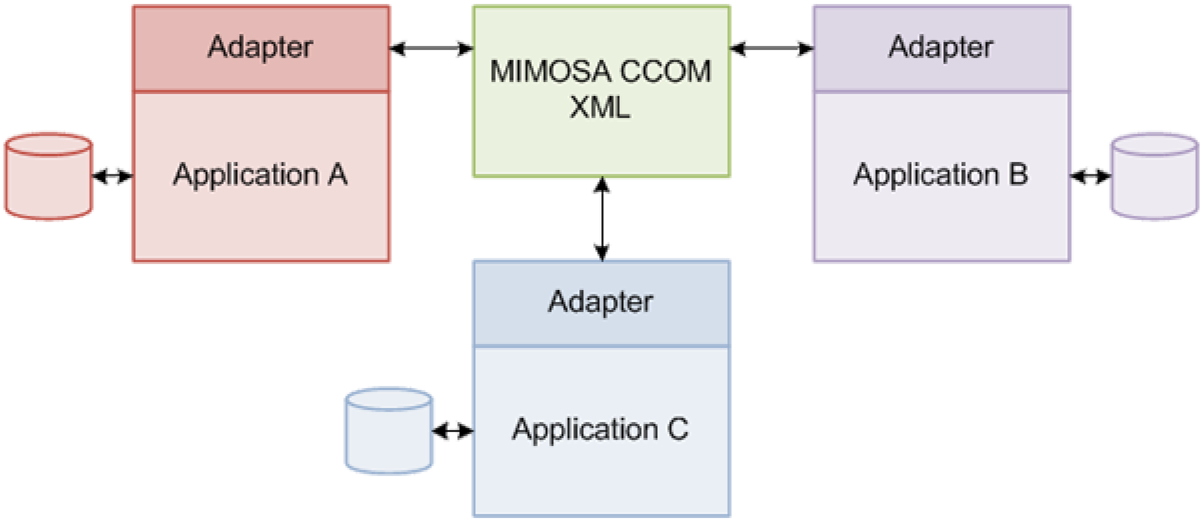


Figure 6 Application(s) will build adapters for exchanging CCOM XML

CCOM is one of the standards included in the OIIE (Open Industrial Interoperability Ecosystem), which brings together several standards and specifications with the aim of supporting broad interoperability between industrial systems in a vendor-neutral fashion.

CCOM provides a common representation of asset information, including:

- As-Engineered and As-Designed engineering data
- As-Built and As-Maintained asset data
- Condition-monitoring, such as asset health, diagnostic, prognostic data
- Asset maintenance and reliability data

CCOM has been implemented in a variety of forms across different operational environments and application integration contexts. The coverage of CCOM can be broken up into several information areas, which are illustrated in Figure 7. The top category, Configuration Management, is the most important as it provides the means to represent the context that much of the other information requires in order to ensure an up-to-date Digital Twin, support advanced analytics, and Condition-Based Maintenance. Another particularly important category is that of Reference Data and Metadata Management, which allows information represented in CCOM to be interpreted correctly by linking the information to standardised reference data sets and tracking provenance in terms of Systems of Record (SoR).

For more details on CCOM, please visit *here*.

© 1998 - 2021 MIMOSA. All rights reserved.

Figure 7 - MIMOSA CCOM Information Scope

CCOM BOD Structure

While CCOM provides an object model and canonical XML representation of that model, the plain XML representation is often unsuitable for more than just basic data integration, particularly when it is necessary to exchange meaningful messages between applications or across an enterprise service bus. The *Open Applications Group Integration Specification* (OAGIS) Business Object Document (BOD) message structure is used to provide additional message concepts that encapsulate a MIMOSA CCOM payload. Specifically, OAGIS Platform Specification 1.2.1 is referenced by CCOM 4.0/4.1. Using the BOD architecture allows consistent message structure and metadata regardless of protocol or data format.

BODs indicate both behavior and structure for messages and the major components of a BOD are depicted below.

© 1998 - 2021 MIMOSA. All rights reserved.

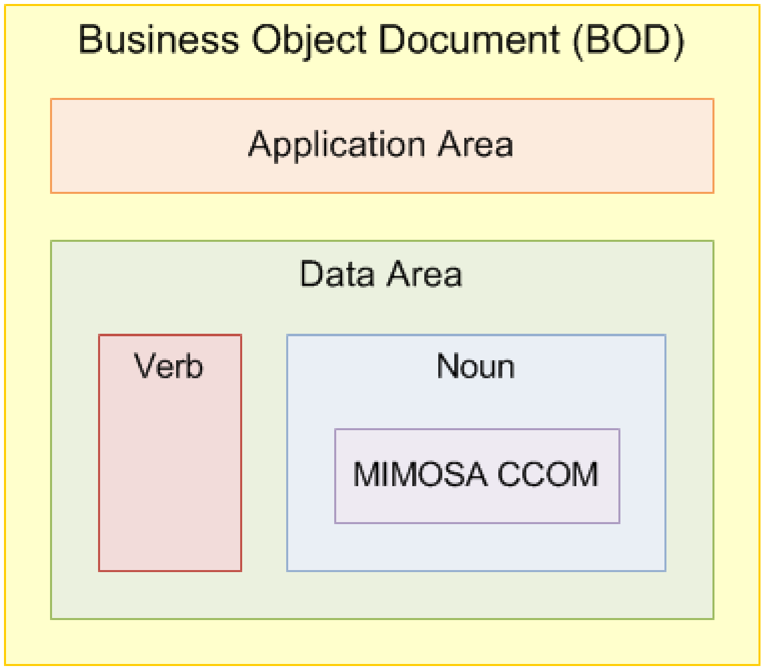


Figure 8 BOD Message Structure

The major components of the BOD are:

- The Application Area is for transactional meta-data, such as identifiers associated with the sender, a creation time stamp, and a unique identifier for the BOD.
- The Data Area contains a Verb (in other words, the operation) as well as a Noun (the business object of that operation).
- The Verb can indicate either what operation to perform (for example, Get) or a response to an operation (for example, Show).
- The Noun instance contains the data representing the business object that is being managed. The Nouns in this case, are composed of MIMOSA CCOM data structures.

All CCOM messages follow the OAGIS BOD (Business Object Document) format and must have a message id, sender id, and message creation timestamp. The message id is a UUID (Universally Unique Identifier) and software frameworks generally include support for UUID generation as part of their base library. The OGI Pilot specifies that UUIDs are also used for sender ids, and that all systems in the environment (those at the least producing MIMOSA CCOM messages) are allocated a UUID. This information is stored in a Register for management and query purposes.

Additionally, in a CCOM message, UUIDs are used to identify all objects within the payload. This is because local object identifiers (e.g. a database primary key) have little meaning to other systems, which in turn use their own local object identifiers. Thus, each system must use shared identifiers when sending CCOM messages.

For more details on CCOM BOD structure, please visit *here*.

MIMOSA SDAIR

The *MIMOSA SDAIR specification* addresses key aspects of OIIE ecosystem administration, with a focus on application-neutral technical and engineering-oriented Master Data Management for life-cycle asset and facilities Management. The SDAIR enables the management of the system of record for systems of record, providing

© 1998 - 2021 MIMOSA. All rights reserved.

among other things, the basis for peer-to-peer based automated information provisioning for all applications that will be part of an OIIE instance. In the loosely coupled peer-to-peer OIIE, the SDAIR plays a significant role in Management of Change, by keeping a full audit log of all individual changes to asset configuration published by each OIIE compliant system. It also enables the management and mapping of Industry Standard Datasheet Definition (ISDD) instances to an owner/operator or supplier's own unique internal standards for datasheet-oriented properties. It accomplishes this in flexible manner, which enables linking to both systems views and multiple breakdown structure views of the plants, platforms and facilities being managed in any given OIIE instance. The systems views and breakdown structures can be imported from one or more designated OIIE-compliant Engineering Design systems of record. It also enables multiple, locally unique identifiers, to be federated to a single true UUID. All information elements, their relationships to each other and their properties are managed and exchanged based on the MIMOSA CCOM information model, which supports a full network model, where each element, defined relationship and their properties are associated with true UUIDs.

For more details on MIMOSA SDAIR specification, please visit *here*.

MIMOSA Service Directory

The Service Directory provides centralized configuration management for the routing of services within an ISBM. An administrator defines applications, the services that each application supports (e.g. publishing information or responding to requests), the scope of the service (e.g. a region or site, a data class such as equipment types, or for an instance of a data class such as a pump), and the corresponding ISBM endpoint, channel, topics, and tokens. An application would then register the services it can provide for a scope with the Service Directory and receive the corresponding ISBM configuration information. Subsequently, applications that need to subscribe or request data would query the Service Directory for a service type and scope and receive the applicable ISBM configuration.

To simplify the configuration management of ISBM adapters, the Service Directory allows centralized channel, topic and security management through an easy-to-use user interface. It also allows a holistic view of the configuration of the ecosystem, allowing users to see the connectivity of applications and errors to be easily spotted. Through the Service Directory, users can easily designate certain applications to be the "service of record" for a given category of information, and this can be enforced through the use of secured channels using token-based credentials.

Systems query the Service Registry providing a service type and scope, and receive back, if authorized, corresponding ISBM channel, topic, and security details. The Service Registry ISBM configuration details are stored as a configuration item for the system's Service Registry Adapter.

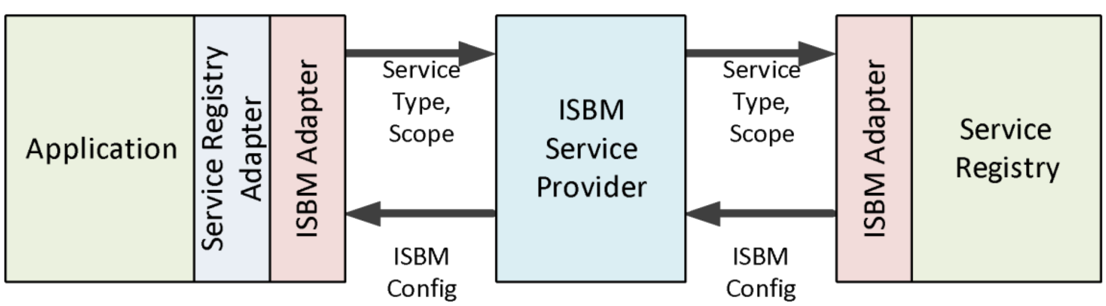


Figure 9 Service Registry Interaction via the ISBM

While the above building blocks provide technical elements to construct an ecosystem, the ecosystem needs to be appropriately managed and governed in order to achieve any effective interoperability. Some of the key requirements for this to occur are:

- Single System of Record (SoR) for any given data class or object to ensure that the authoritative source for the class/object is known. This relationship may be particular to a certain scope, for example, to a certain region or site (e.g. SoR of data sheets for North America), data class (e.g. SoR of data sheets for pumps) or specific data object (e.g. SoR for data sheets for P-1000).

© 1998 - 2021 MIMOSA. All rights reserved.

- Single Request Service Provider for a given scope to ensure against multiple service providers producing conflicting information.
- Supplier separation between key platforms to ensure suppliers do not try to subsume other application areas and subvert the ecosystem through the introduction of proprietary connections or performance manipulation.
- Appropriate caching to limit the number of requests to the ws-CIR and Service Registry and appropriate cache flushing management applied when changing identifiers or services.

For more details on MIMOSA Service Directory, please visit *here*.

MIMOSA ISDD Project

The Industry Standard Datasheet Definitions (ISDD) Project is developing a set of digital data sheets for key classes of equipment based on Industry Standard Datasheets: that is, data sheets published by authoritative industry bodies and associations such as ISA, API, and PIP. Each ISDD standardizes a set of engineering, asset, and/or product model properties in a form that is reusable, mappable, supplier-neutral, and fully machine interpretable. This results in improvements across the entire asset lifecycle management: for example, the exchange of functional requirements with manufacturers that can then be used in automated make/model match-up processes, the sharing of detailed product data relating to installed assets that improves the long-term operation and maintenance of those assets.

For more details on MIMOSA ISDD project, please visit *here*.

OpenO&M ISBM

The ISBM forms the backbone of the OIIE as web-service based Enterprise Services Bus. The predominant communication method in an OIIE instance is via an ISBM, which supports both request-response and publish-subscribe communication modalities. The ISBM is composed of a set of service interfaces to support various functionality such as configuration and management, request-response communication, publish-subscribe communication, and push notifications to client systems. The complete set of services is as follows:

- Channel Management Service: channel and security configuration
- Notification Service: allow notifications of new messages
- Provider Publication Service: publish messages
- Consumer Publication Service: read published messages
- Provider Request Service: read and respond to requests
- Consumer Request Service: send requests and read responses
- Configuration Discovery Service: provide service providers supported feature set

The ISBM 2.0 specification defines a SOAP Web Service implementation of the ISA 95.00.06 Messaging Service Model (MSM) as well as describe a plain HTTP/JSON REST interface defined by the OpenO&M ISBM Joint Working Group (JWG). The features and capabilities of the latest ISBM specification were validated through the *OIIE OGI Pilot* to ensure that it is fit-for-purpose.

The specification defines a minimal interface subset to message exchange middleware using standard Web Service and REST interfaces. Publish-subscribe and request-response messaging patterns are supported through a consistent and unified model. Message routing is conducted through shared channels and topics, and optionally, XPath/JSONPath filtering for granular content-based filtering. An asynchronous Web Service callback or an asynchronous callback REST service is also provided to clients for notification of received messages.

© 1998 - 2021 MIMOSA. All rights reserved.

Token-based security for channels is specified to support multiple authorization models, from basic credential exchange to federated identity providers.

The benefit of ISBM implementation specification is that it allows applications to expose a single, standardized interface instead of a custom-built interface for every version and format of message exchange systems. It also allows applications to select REST or Web Services based on the application requirements. The goal is to further interoperability in application to application communications.

For more details on OpenO&M ISBM, please visit *here*.

OpenO&M CIR

The purpose of the OpenO&M Common Interoperability Register specification is to provide a standards-based, vendor-neutral specification for a Web Service interface for interactions with an object identification registry. This supports the harmonization and standardized lookup of locally-unique identifiers for an identical object (including data dictionary classifications and attributes) used in multiple information systems. Each system typically maintains its own set of identifiers for its objects. For example, System A may use a set of auto-incrementing integers; System B may use strings; System C may use UUIDs. As the objects in each of these systems may be the same business object (albeit instantiated in three different systems), in order to link the three objects, the CIR is used to define an overarching identifier.

Another common case is for an application interface adapter mapping between internal application identifiers and the data exchange scheme (e.g. MIMOSA CCOM). In this case, the ws-CIR would register two objects – one for the application and one for the data exchange scheme. The ws-CIR would store the identifier for each object, for example, an integer against the object in the application and a UUID for the object in MIMOSA CCOM. This then allows two applications to communicate with each other using shared identifiers which are then translatable to objects within their own internal application.

Each OIIE compliant system shall utilize the CIR when creating new objects in their local system, associating their locally-unique identifier with the CIRID. Once all systems have registered their objects with the CIR and an equivalency matching process has been conducted, the CIR can be used to translate between identifiers. The Transform Engine uses the CIR to map identifiers to/from ISO 15926 to the CIRID. The best practice for MIMOSA CCOM-compliant registry systems is to natively utilize the CIRID UUID as the identification of MIMOSA CCOM objects.

For more details on OpenO&M CIR, please visit *here*.

Building an OIIE/ISBM Adaptor

The mechanics of data exchange involve provider systems exporting data to a format, transportation of that data, and consuming systems importing the data. The format conversion and use of transport typically involve the use of “adaptors” – application modules that provide the ability to format data for import/export and support system to system communications. Each participant that needs to interact with the OIIE ecosystem is responsible for the development of these adapters.

The analysis and identification process of functional requirements for an OIIE/ISBM adaptor may leverage material such as the OIIE Use Cases, Scenarios and Events. The functional requirements and subsequent design will help indicate which sections of the ISBM specification are relevant for an application. Typical requirement and design decisions that will be encountered are:

- What role does my application play?
- Which transaction models will my application need to support?
- Which channel management WSDL methods need to be supported?

© 1998 - 2021 MIMOSA. All rights reserved.

- Will my application be able to host a Web Service for asynchronous callback or will I need to revert to polling?
- What events will trigger a payload to be sent on the ISBM? What events will be triggered when a message is received?
- How will the ISBM module be configured including channel, topic and token configuration?
- How are other configuration items, such as polling or retry intervals, configured?
- How are channels and topics persisted across application restarts?
- Where will ISBM activity be logged and how are errors presented to users?
- What ISBM activity needs to be persisted for audit purposes?

The steps involved before actually implementing an OIIE/ISBM adaptor are provided on *MIMOSA website*, which includes first identifying in which Use Cases, Scenarios and Events the system needs to participate. Then, identify the underlying specific CCOM elements and perform the mappings to those CCOM elements. The relevant BODs should be identified for transferring the payloads to specific systems. An organization's role will determine the specific interfaces of the ISBM they need to build adaptor for. For example, an organization may just need to build an adaptor to consume the publications posted by another provider organization. The final part will be connecting to specific component(s) of the OIIE ecosystem via ISBM based on their requirements.

Document Versioning

Version	Date	Major Changes
1.0	2020-07-22	Initial document

© 1998 - 2021 MIMOSA. All rights reserved.