# MIMOSA and OPC UA Applied in Wapice Remote Management System

*Mickey Shroff, Jyrki Keskinen, Oskar Norrback, Sakari Junnila, Jaakko Takaluoma, Pasi Tuominen*
Wapice Oy, Yliopistonranta 5 (3. krs), 65200 Vaasa
Tel. +358 6 319 4000, Fax + 358 6 319 4001, firstname.lastname[at]wapice.com, http://www.wapice.com

Keywords: MIMOSA, OPC UA, distributed, remote monitoring, SOA

## ABSTRACT

This paper presents our experiences on implementing a remote monitoring solution utilizing OPC UA and some characteristics of the MIMOSA standards /1/3/. The implemented system is highly distributed and the effects of distribution on the node's response time and communication throughput were investigated. Our focus is on the implementation of the following functionalities: data acquisition methods, data structures for temporary storage, data manipulation and storage routines and rule engines. The goal was to maintain compatibility and interoperability with other systems.

MIMOSA relies on ISO-13374 standard that defines architecture of data processing functionality blocks. In performance critical systems, OPC UA was found a suitable solution for transferring automation and control boundary data to higher levels in a structured manner. It was found that the placement of data processing logic is essential for well performing distributed system. Distribution was also found beneficial when business critical data processing algorithms were provided by 3rd parties.

## 1 INTRODUCTION

Distribution of computing and resources is important in modern automation systems. Distributed systems are generally less vulnerable and more robust than centralized systems. They can transfer load between the nodes, depending on each node's computing, communication, memory and power resources. The task distribution can be reconfigured during the operation when new nodes are added, existing ones removed or when nodes properties change. To effectively utilize distribution in large scale systems and to integrate the different devices it is important to use standardized architecture solutions and ontology.

This paper studies the use of MIMOSA trade association specifications and the applicability of OPC Unified Architecture (OPC UA) to implement a standardized distributed environment. MIMOSA is a non-profit organization that is part of the Open O&M (Operations and Maintenance) initiative. The purpose of the initiative is to encourage usage of open information standards when implementing systems for O&M in the manufacturing, fleet and facility environments. MIMOSA's main publications are the Open System Architecture for Condition-Based Maintenance (OSA-CBM) and Open System Architecture for Enterprise Application Integration (OSA-EAI) specifications. /3/4/5/

The OSA-CBM specification defines architecture for moving information in a Condition-Based Maintenance (CBM) system. CBM refers to maintenance that is done when a need arises based on condition monitoring done to the target system. This is different compared to planned maintenance where maintenance is executed at a certain interval (running hours or similar) regardless of the equipments health (car maintenance every x kilometers for example). OSA-CBM is an implementation of the ISO-13374 standard, which defines the six blocks of functionality in a condition monitoring system and the general inputs and outputs of those blocks /2/. OSA-CBM additionally defines data structures and interface methods for the communication between the functionality blocks. /4/

Open System Architecture for Enterprise Application Integration (OSA-EAI) is architecture for integrating enterprises different systems (engineering, maintenance, operations etc.) with each other. /5/

The next chapter presents the MIMOSA and OPC UA concepts. In Chapter 3 it is shown how MIMOSA and OPC UA were applied in Wapice Remote Management platform. The results of our study are presented in Chapter 4. Finally, conclusions are drawn.

## 2 MIMOSA AND OPC UA CONCEPTS

The MIMOSA specifications define ontology for the target environment. This is to ensure that professionals working in the same field can discuss issues using commonly agreed terms and definitions.

Based on this ontology MIMOSA also supplies a Common Conceptual Object Model (CCOM) as a Unified Modeling Language (UML) diagram. Furthermore, a complete relational database schema based on the CCOM is supplied in the OSA-EAI specification.

Interfaces between different components are defined as Web Service Definition Language (WSDL) schemas. The data transferred between the components are defined with eXtensible Markup Language (XML) schemas. /5/

The ISO-13374 standard defines six independent blocks of functionality in a condition monitoring system: /2/

- **Data Acquisition (DA):** Converts the sensor output to a digital value representing a physical quantity and information related to the specific measurement, such as the time, calibration, data quality etc.
- **Data Manipulation (DM):** Performs signal analysis and calculates virtual measurements based on raw data.
- **State Detection (SD):** Management of different “profiles”, which can be used to detect abnormalities when new measurements are read. SD can be used to generate alerts or alarms when certain states occur.
- **Health Assessment (HA):** Diagnoses faults, provides health assessment for the current state of the equipment.
- **Prognostics Assessment (PA):** Generates a prognosis for future health states, failure modes and remaining lifetime based on the current health assessment and projected usage load.
- **Advisory Generation (AG):** Suggests actions (maintenance or operational changes) required to optimize the life of the process and/or equipment.

In our implementation, we have focused on implementing functionalities of the first three functional blocks, according to Figure 1.

MIMOSA offers four advantages gained from implementing their OSA-CBM standard: /4/

- **Cost** – Cost savings can be made because the software developers will not have to spend time designing new ways of transferring the data within the system, essentially reinventing the wheel. Multiple software vendors can also be used since the OSA-CBM standard is modular, each vendor working on their own.
- **Specialization** – Since vendors don’t have to commit to providing entire CBM systems, they can choose to specialize in refining the functionality of a specific CBM block. This also gives the opportunity for smaller companies that cannot afford to develop full CBM systems to compete with large companies.
- **Competition** – Since all compliant vendors are using the same input/output interfaces direct comparisons can be made on functional level instead of comparing systems on total solution or system level.
- **Cooperation** – In addition to competition, OSA-CBM can also increase cooperation. Since the standard defines the interfaces CBM modules developed by different vendors will be able to communicate with the others seamlessly if they were implemented using the same technologies.

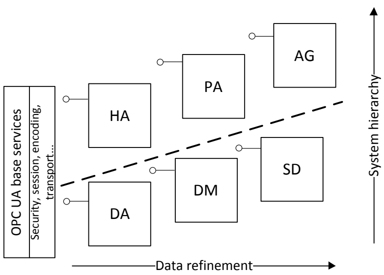



**Figure 1.** MIMOSA and OPC UA integration and system hierarchies.

OPC UA was developed as the next generation integration standard by the OPC Foundation. OPC UA addresses deficiencies of the classical OPC specifications by defining e.g. a uniform address space model, support for custom information models and platform independence. OPC UA security features support message encryption and most importantly endpoint authentication. In the world of Stuxnet /9/ and its derivatives, it's important to secure point-to-point communication, authenticate communicating identities and reject unauthorized actions.

OPC UA brings the Service-oriented architecture (SOA), commonly used in enterprise application integration, to automation and control systems domain (Figure 1). OPC UA enables a much wider variety of application domains to adopt the technology. Now, everything from embedded platforms with limited resources to large scale enterprise systems independent of the programming language and the platform they are deployed on may be introduced to the new integration standard. /1/10/

# 3 APPLYING MIMOSA AND OPC UA IN WAPICE REMOTE MANAGEMENT PLATFORM

Wapice Remote Management (WRM) platform design was inspired by MIMOSA to some degree, but the specifications have not been implemented fully. Terms from the OSA-EAI terminology dictionary and simplified parts of the CCOM and Common Relational Information Schema (CRIS) specifications have been used when designing the WRM object model and database schema. Communication between WRM devices and the backend server is implemented using web services and it could be made MIMOSA compliant if needed.

MIMOSA's CCOM object model and CRIS database schemas were not used directly, since after our own initial research and reviewing opinions from other sources it was agreed that the database model was too complex for our use case. CRIS was created to be very flexible with minimal hardcoded columns in the tables. It also has the possibility to add solution specific columns or object properties via generic custom type tables and foreign keys. Unfortunately, this generality comes at a performance cost, resulting in complex queries and storage overhead. /6/

OPC UA implements the Data Access (DA) information model. The DA information model is described in OPC UA specification part 8 /8/. Clients utilize needed service profiles to support DA model. The DataAccess Client Facet is important part of implementing DA. UA Profiles provide means for the clients and servers not to be concerned about UA client and server implementation variations. Profiles are described in UA specification part 7 /7/. Basically The UA DA module describes how analog and digital data can be presented in UA context. Data is accessed using standard UA read or subscription mechanisms.

Figure 2 depicts how functionality blocks defined in ISO-13374 /2/ can be distributed and deployed in WRM platform. Generic products that can support many different DA blocks require modular approach. Each DA block can acquire its data behind different physical connections (ZigBee, CAN, OPC UA, FTP etc.).

Low level measurements (like sensors) support only DA block functionality and there is no local data storage. Fieldbus equipments can implement DA and DM block functionalities. For example relays have internal disturbance records containing very accurate data. Automation systems (AS) can implement DA, DM, and SD block functionalities. AS can contain specific analyzing functionalities such as 3rd party commercial tools.

Automation and control system domain (ACD) can provide refined data (e.g. FFT processed data) or raw process values cyclically or event-based to WRM terminal or other similar device.

Figure 3 depicts data flows in the WRM platform. Data sampling acquires data from DA blocks. A cyclical process moves sampled data to cache where data is available for filtering and statistical calculations. Processed data is stored into persistent data storage.

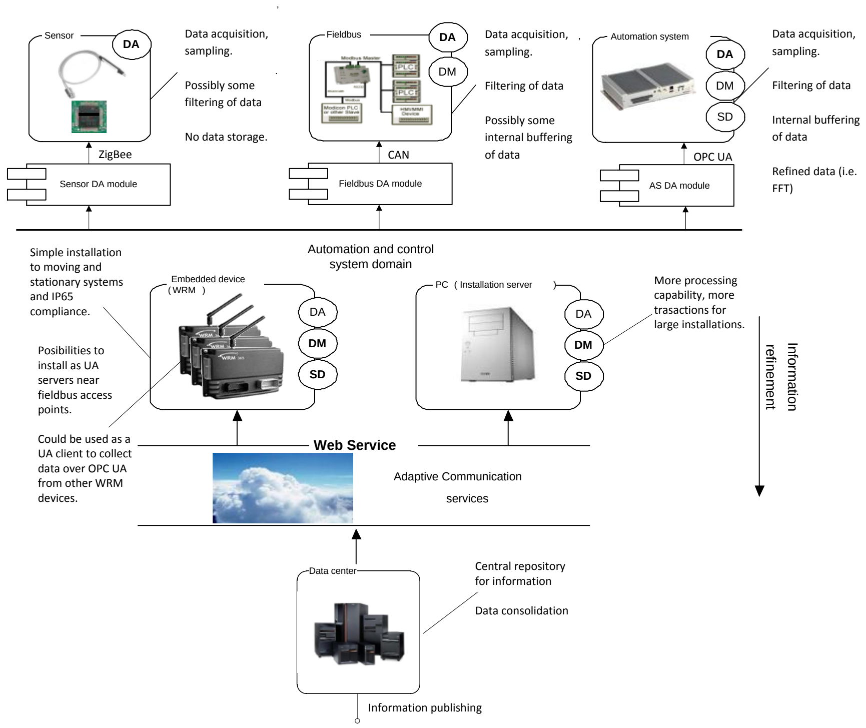



**Figure 2.** WRM Remote Management Platform.

## 3.1 Data Sampling

In WRM, sampling occurs in sensory, fieldbus and OPC UA boundaries. Real-time sampling is difficult in a generic system. There is always some delay and typical implementations of remote monitoring and management systems do not have strict real-time requirements comparable to Programmable Logic Controller (PLC) or Supervisory Control and Data Acquisition (SCADA) based systems. Delays are mainly caused by latency in communication, data storage (writing/reading), data conversions and applying related information (such as time, calibration, and data quality)

The real-time characteristics of acquired data depend on the level of hierarchy at which the data is read and where the data is used. In sensory and fieldbus levels the data acquisition sampling rates are relatively high compared to OPC DA access points and even more so at the Enterprise Resource Planning (ERP) boundaries. Also, at the fieldbus level it’s possible to define different sampling scenarios.

In WRM, data sampling is divided into three categories: normal sampling, event-based actions and data-pass-through. Normal sampling is used to acquire cyclical data from a specific DA block. Event-based actions are executed when specific conditions are true. These conditions are evaluated by the rule engine. For example an event-base action can fetch data from a relay using ftp. When data is passed through no data sampling is used and device acts like a gateway.

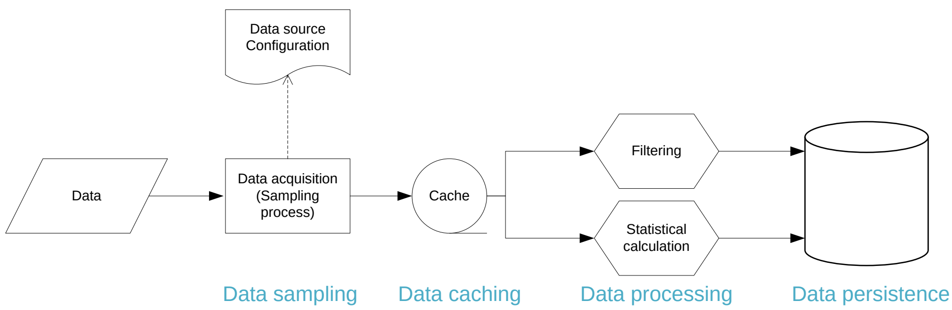



**Figure 3.** Data sampling and processing.

## 3.2 Data processing and storage routines

Usually there is no need to store all sampled data to database. In the WRM system, data can be filtered before storing persistently. Main tools for that are parameterized filters (e.g. deadbanding) and statistical operations (e.g. min, max and average). Filtering can also be controlled by rule engine. Normally data is filtered before persistence but when a rule triggers the data can be stored without filtering.

Ring buffer (cache) is used in WRM system when only required amount of data is kept in memory. The buffering automatically preserves memory by evicting old values. Buffers allow real-time timeframe processing of data. The amount of slots maps to time.

Data persistency can be implemented using sequential file or standard database storage. Sequential files are fast, they offer no or very little indexing capabilities. Indexing must be implemented separately (directory structure file naming, some index structure). Inserts to sequential files are fast; because new data can be added to the end of the file. Fetching data from sequential files are fairly slow, but this is typically non-important defect since the file is fairly seldom or never scanned. Databases bring concurrency control and often very powerful indexing system into use. Indexing slows down the insertion of data; when writing new data to the disk the index or even many indexes must be updated also. However, databases provide very fast access to random data. That is the power of indexes.

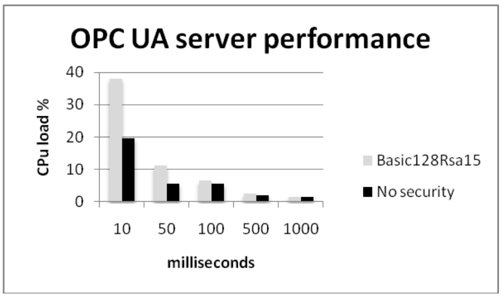
# 4 RESULTS

It was observed that web services caused unnecessary overhead when DA, DM and SD blocks were located in the same device. In this case, for performance reasons connections between DA, DM, and SD modules are implemented without web services utilizing interprocess calls. In large scale systems, WRM can be distributed by deploying multiple ACDs (figure 2). In that case, every ACD is an independent unit and contains its own definitions for DA, DM and SD modules. Distribution was also found beneficial in instances were business critical “secret” data processing algorithms are provided by e.g. 3rd party system. For redundancy reasons, the most important information can be secured using duplicated DA sources.

It was observer that in a large systems the amount of data can grow to even terabytes quite quickly. Consider a large process like power plant with couple of engines (5-10). The amount of measurement points can increase rapidly. It would be nice to get all the data from monitored systems, but it’s a fact that the data amount must be limited due to network and storage constraints. The WRM system limits the amount of data using statistical calculation and intelligent rule-based filters.

At the ACD boundary database servers were not utilized at all. Instead of using sequential files, light embedded database was observed to be an optimal choice. This decision provided the best balance between performance and flexibility compared to sequential files and standard database solutions.

Figure 4 depicts OPC UA server and client performance measured in WRM 247 /11/ device. In both graphs, data was read using DA interface’s AnalogItem subscription service, where the process value size was 32 bits. Server performance was measured by observing the effects of OPC UA subscription interval on the CPU load. The UA client performance was measured using various client connection amounts. It can be seen that encrypting the communication channel did not cause significant CPU load increase on the OPC UA server machine, when the subscription interval of the OPC UA server was greater than 100 ms. One OPC UA client could collect data from many OPC UA servers at the same time. The CPU load of the OPC UA client machine was observed to increase gracefully, in linear manner, in perspective to the number of OPC UA servers the client was connected to.



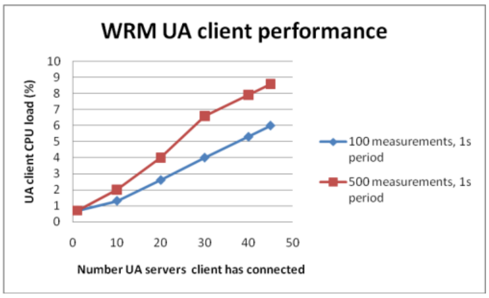




**Figure 4.** OPC UA server and client performance.

## 5 CONCLUSIONS

Even though the MIMOSA offers tempting advantages we decided not to implement the specifications fully mostly due to performance reasons. Our inspiration for the design was taken from the Terminology Dictionary, CCOM object model and CRIS database specifications.

OPC UA was found a suitable solution for transferring automation and control boundary data to higher levels in a structured manner (DA module). OPC UA supports distribution of automation elements. UA server can be deployed in embedded devices and PCs. We used UA servers to collect sensory and fieldbus data. Embedded UA client was used to collect data from many UA servers and provide a single point of access otherwise to a highly distributed automation system.

Effects of distribution in perspective of performance were evaluated. Latency variations were not that much of importance compared to accurate time stamping. It was found that the amount of data generated at the automation and control levels should be minimized as early as possible to allow more information to flow to higher system levels. This was accomplished with doing data filtering and statistical calculations at the nearest available data access point.

Future work could involve implementing the functionality of the Health Assessment (HA), Prognostics Assessment (PA) and Advisory Generation (AG) blocks to WRM.

## 6 REFERENCES


1 Hannelius T., Shroff M., Tuominen P.: Embedding OPC Unified Architecture, Automaatio XVIII seminar 2009, ISBN 978-952-5183-35-1.

2 International Organization for Standardization, Geneva, Switzerland, ISO 13374-1:2003: Condition monitoring and diagnostics of machines - Data processing, communication and presentation - Part 1: General guidelines, 2003.

3 MIMOSA: What is MIMOSA?, available at: http://www.mimosa.org/?q=about/what-mimosa, [referenced 2011-02-17]

4 MIMOSA: OSA-CBM specification, version 3.3.0, available at: http://www.mimosa.org/?q=resources/specs/osa-cbm-v330

5 MIMOSA: OSA-EAI specification, version 3.2.2, available at: http://www.mimosa.org/?q=resources/specs/osa-eai-v322

6 Mathew, Avin D., Zhang, Liqun, Zhang, Sheng, & Ma, Lin (2006) A review of the MIMOSA OSA-EAI database for condition monitoring systems. In: World Congress on Engineering Asset Management, 11-14 July 2006, Gold Coast, Australia.

7 OPC Foundation: OPC Unified Architecture Specification Part 7: Profiles, Release 1.00, February 2009

8 OPC Foundation: OPC Unified Architecture Specification Part 8: Data Access 1.01 Specification, Release 1.01, February 2009

9 Symantec, Security Response, W32.Stuxnet Dossier, available at: http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf [referenced 2011-02-16]

10 Tuomi, A., "Application integration for condition based maintenance," Master Thesis, Aalto University School of Science and Technology, Faculty of Electronics, Communications and Automation Espoo, Finland, 12.5.2010.

11 WRM 247 Technical Data, available at: http://www.wapice.com/wapice_cms/files/WRM_247_rgb_full.pdf, [referenced 2011-02-18]