Information Modeling for Manufacturing Intelligence

by

Paul Hunkar

Subramanian (Subbu) Periaswamy

Keywords: OPC UA, ISA-95, Manufacturing Intelligence, Unified Architecture, IEC 62264, IEC 62541,

INTRODUCTION

According to Wikipedia: "**Enterprise manufacturing intelligence (EMI),** or simply manufacturing intelligence (MI), is a term which applies to software used to bring a corporation's manufacturing-related data together from many sources for the purposes of reporting, analysis, visual summaries, and passing data between enterprise-level and plant-floor systems (see Figure 1). As data is combined from multiple sources, it can be given a new structure or context that will help users find what they need regardless of where it came from. The primary goal is to turn large amounts of manufacturing data into real knowledge and drive business results based on that knowledge."

Manufacturing intelligence is attained by an organization by achieving consistency in operations and improved productivity. In today's world information is provided from many sources, each with their own data representation. The key challenge is to make sure that the data is generic based on standards but also that the data is specific enough to meet the organizations requirements.

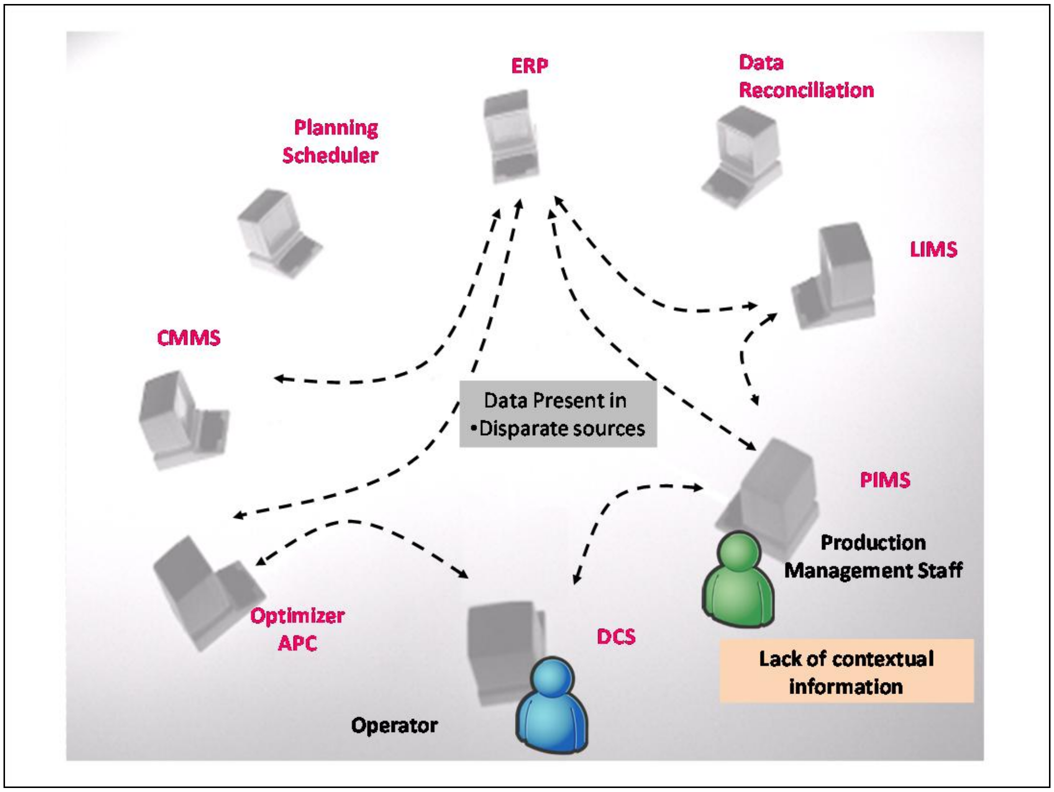


Figure 1 Current Information and Data Sources

This paper illustrates the additional value that can result from using the ISA 95 OPC UA companion specification [8] as part of a complete solution. The companion specification can be the base of the manufacturing information model and uses the flexibility of OPC UA to build a custom information model specific to the organization. This information model can include custom KPIs to improve the visibility of the System and help with continuous improvement, together achieving Manufacturing Intelligence.

- Associate data in different systems with a context.
- Translate data to contextual information.
- Present meaningful information to people of different roles in different ways.
- Help decision makers take informed decisions
- Enabled planners to have a real-time visibility of the production
- Assist the Maintenance / Operations teams to related information from all the other systems. E.g. Maintenance team may find it useful to have the past event history of an equipment from an asset monitoring system and its related diagnostic procedures from a procedure system.

CHALLENGES IN BUILDING A SYSTEM

A developer encounters a number of challenges when building custom systems to manage numerous data source and provide valuable KPIs. This section will illustrate a number of common concerns

CHALLENGE - WHAT YOU NEED IS NOT WHAT I NEED

All users are not the same, the information and the level of it required by different users is usually different. Additional wrinkles thrown into this scenario is that systems from different vendors are clubbed together to meet the user's requirements for the business. This can introduce a huge disparity in the data since each vendor may provide the data in different manners. One user might be more interested in the accounting aspect of the data, while another user might be more interested in the visibility of the process. The different user requirements coupled with the integration of the data from disparate sources, all into an intuitive interface which provides real time visibility but needs to be achieved for a low cost is a highly difficult to achieve.

CHALLENGE - HOW DO I PUT SOMETHING SIMILAR AT MY OTHER PLANT?

The user request is complicated. The key items being

- Integrate data from different systems
- Present specific KPIs

What matters most for the user to make informed decisions and improve visibility of the process throughout the organization? The manner of calculating KPIs may be based on a standard template but it may be customized or improved based on specialized knowledge the user acquired through years of business experience. So there is a requirement to build the KPI using a custom complex formula. Once everything is done, after a whooping sum of effort, the user asks for something similar at his other site. Please note that it is not the same but similar, which means many of the KPIs and their meanings might be different. Had the solution been based on a custom implementation then the entire effort may have to be recreated. This results in another round of data integration from different groups of systems and another custom set of formula to be implemented and shown in the unique way the user wants to see it. Sounds like lot of work and is there a way to reuse things?

CHALLENGE – HOW DO I IMPROVE EFFECTIVENESS

Real time visibility of the Key Performance Indicators KPIs is crucial for the improvement of productivity and also for adapting to changes when the business demands it. KPIs are the key items which help the decision makers take crucial decisions ensuring productivity is high and goals are achieved. But KPIs are not that easy to build and track, let alone identify.

DYNAMIC INFORMATION MODELING AND KPI CALCULATION: VALUE ADDITION

Every organization has a unique way of looking at information. They utilize their specialized domain knowledge. So KPIs needs to be calculated and built in an organization specific manner. Implementation of a special application for each of the customer is time and effort consuming. Using the information modeling of OPC UA to build relevant data models and using custom calculation scripts for pulling data from the information model and creating KPIs and populating them in the information model is the easiest way to implement a KPI solution for organizations (see Figure 2). OPC UA serves as a platform which relies more on configuration than implementation thereby adding value to the vendors by provide relevant information such as KPIs for the customers. This customized solution platform enabling specific information to be shown for the customers is a major value addition for them too.

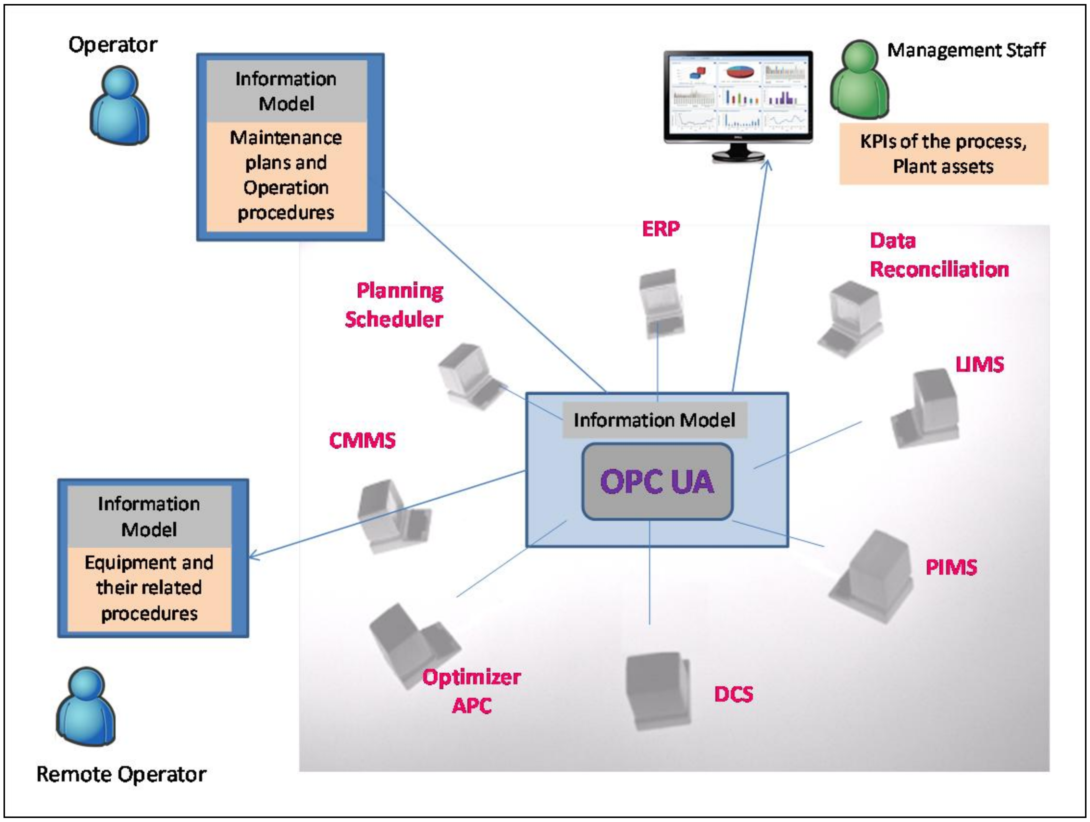


Figure 2 - OPC UA Information Model

AN OVERVIEW OF OPC UA AND INFORMATION MODELS

CONCEPTS

OPC UA (also known as IEC 62541) provides a framework that can be used to represent complex information as Objects in an address space which can be accessed with standard web services. It supports highly secure communication over multiple communication technologies and can be implement in any of a number of programming languages on any platform.

[Authors Note: The Overview of OPC UA and Information Models section in this paper is an edited version of what can be found in the listed OPC UA specifications or in the ISA-95 OPC UA Companion Specification. It is included here with the permission of the OPC Foundation and the editors of these specifications. If additional details are desired please review these documents [3],[4],[5],[6],[7]].

These Objects consist of Nodes connected by References. Different classes of Nodes convey different semantics. For example a Variable Node represents a value that can be read or written. The Variable Node has an associated DataType that can define the actual value, such as a string, float, structure etc. It can also describe the variable value as a variant. A Method Node represents a function that can be called. Every Node has a number of Attributes including a unique identifier called a NodeId and non-localized name called as BrowseName. An Object representing a 'Reservation' is shown in Figure 3.

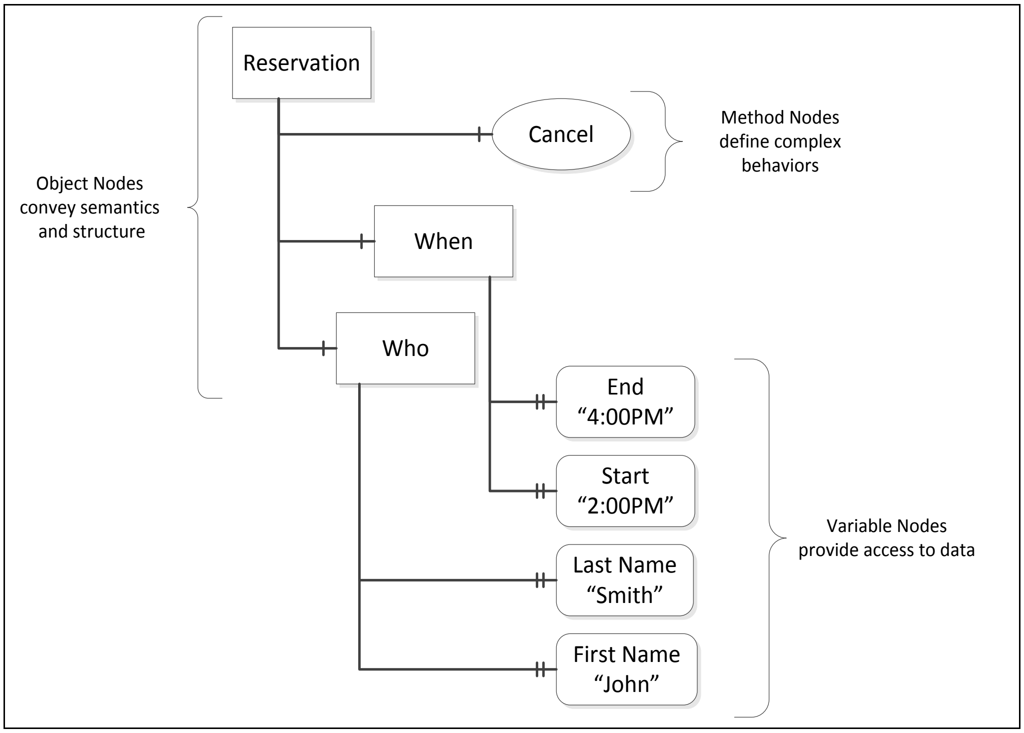


Figure 3 – A Basic Object in an OPC UA Address Space

Object and Variable Nodes are called Instance Nodes and they always reference a Type Definition (ObjectType or VariableType) Node which describes their semantics and structure. Figure 4 illustrates the relationship between an Instance and its Type Definition.

The Type Nodes are templates that define all of the children that can be present in an Instance of the Type. In the example in Figure 4 the PersonType ObjectType defines two children: First Name and Last Name. All instances of PersonType are expected to have the same children with the same BrowseNames. Within a Type the BrowseNames uniquely identify the child. This means Client applications can be designed to search for children based on the BrowseNames from the Type instead of NodeIds. This eliminates the need for manual reconfiguration of systems if a Client uses Types that multiple devices implement.

OPC UA also supports the concept of sub typing. This allows a modeler to take an existing Type and extend it. There are rules regarding sub typing defined in OPC UA Part 3[5], but in general they allow the extension of a given type or the restriction of a DataType. For example the modeler may decide that the existing ObjectType in some cases needs an additional Variable. The modeler can create a Subtype of the ObjectType and add the Variable. A client that is expecting the parent ObjectType can treat the new ObjectType as if it was of the parent ObjectType. With regard to DataTypes, if a Variable is defined to have a numeric value, a sub type could restrict the value to a float.

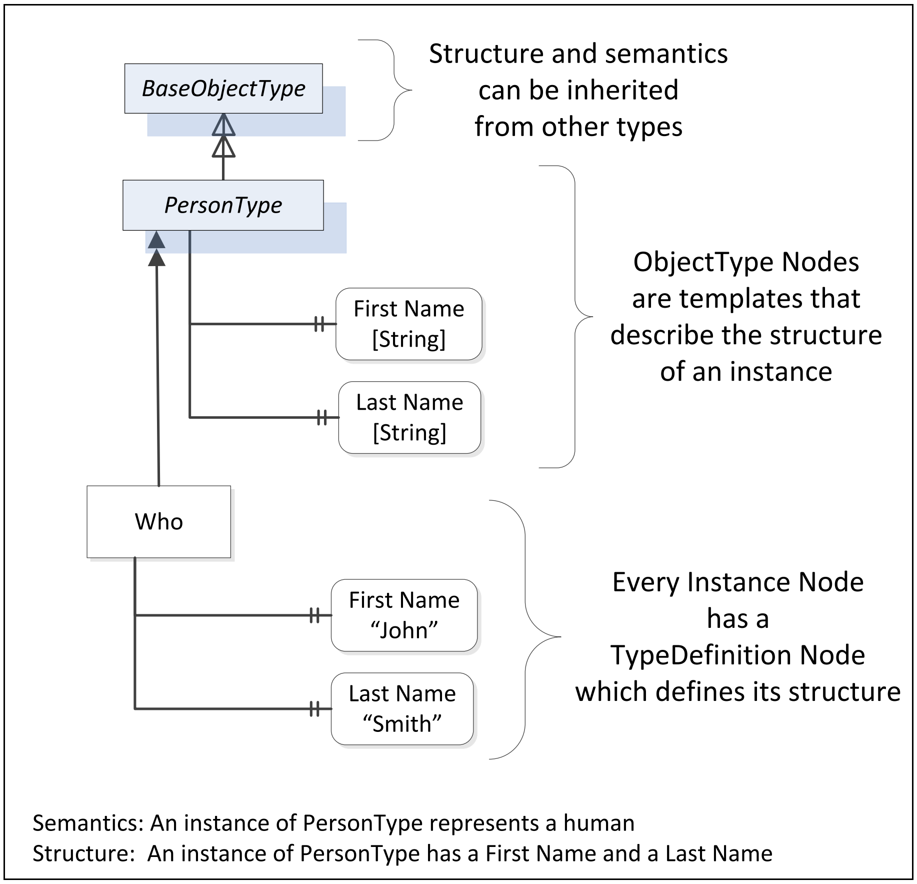


Figure 4 – The Relationship between Type Definitions and Instances

References allow Nodes to be connected together in ways that describe their relationships. All References have a ReferenceType that specifies the semantics of the relationship. References can be hierarchical or non-hierarchical. Hierarchical references are used to create the structure of Objects and Variables. Non-hierarchical references are used to create arbitrary associations. Applications can define their own ReferenceType by creating a Subtype of an existing ReferenceType. Subtypes inherit the semantics of the parent but may add additional restrictions. Figure 5 depicts several references connecting different Objects.

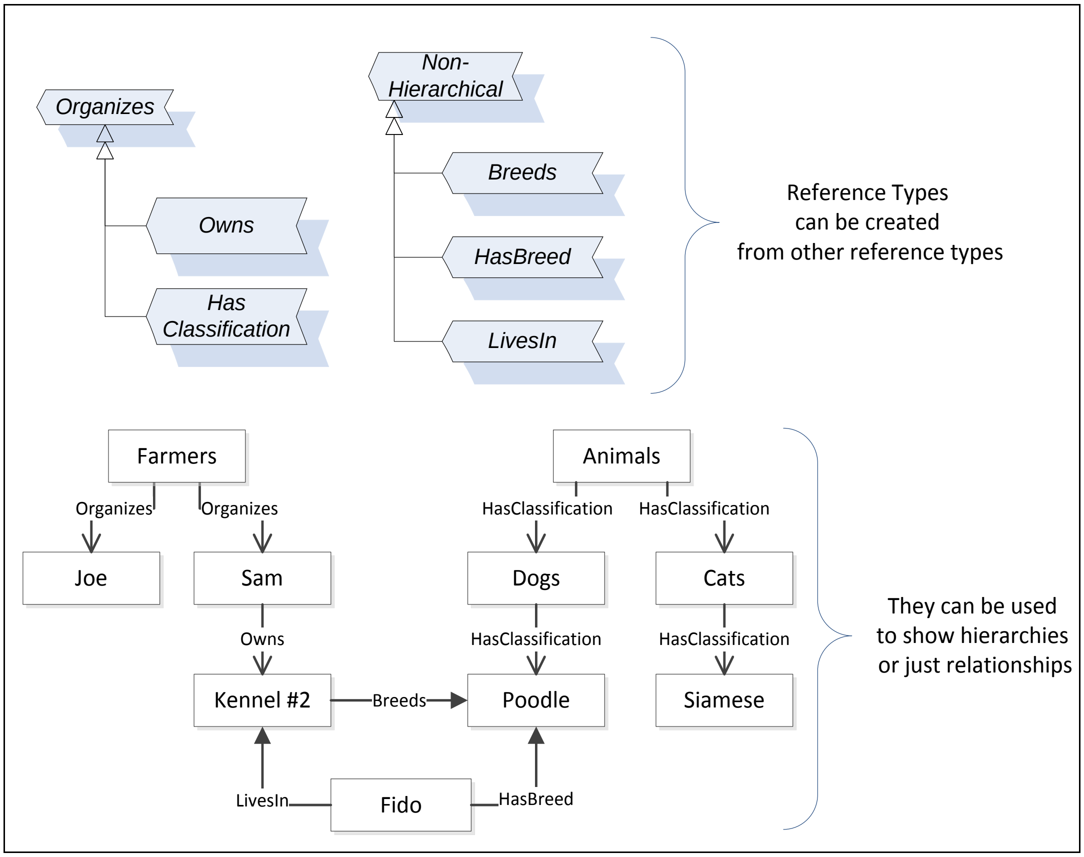


Figure 5 – Examples of References between Objects

The figures above use a notation that was developed for the OPC UA specification. The notation is summarized in Figure 6. UML representations can also be used; however, the OPC UA notation is less ambiguous because there is a direct mapping from the elements in the figures to Nodes in the address space of an OPC UA server.

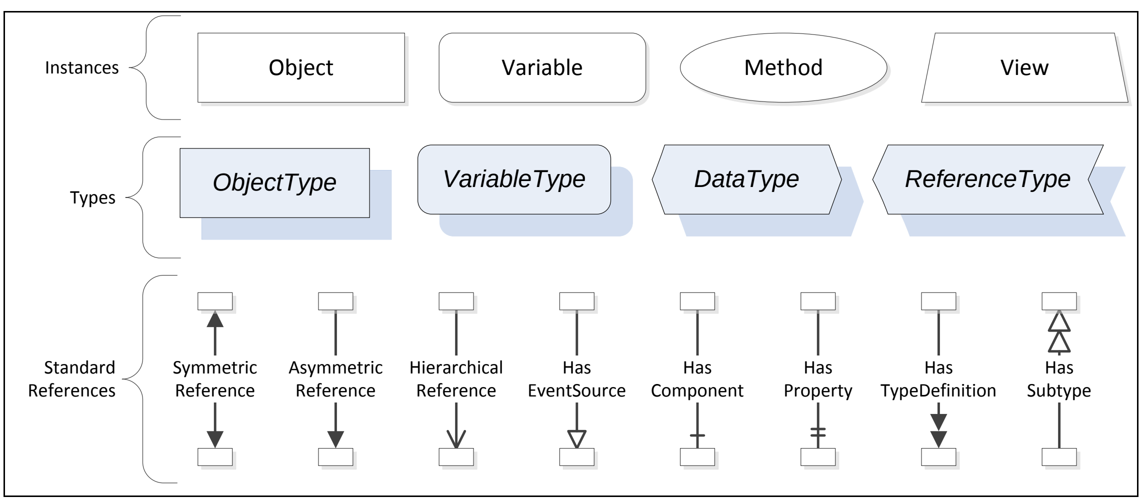


Figure 6 - The OPC UA Information Model Notation

A complete description of the different types of Nodes and References can be found in [7] and the base OPC UA Address space is described in [5].

OPC UA specification defines a very wide range of functionality in its basic information model. It is not expect that all clients or servers support all functionality in the OPC UA specifications. OPC UA includes the concept of profiles, which segment the functionality into testable certifiable units. This allows the development of companion specification (such as OPC UA for ISA-95 [8]) that can describe the subset of functionality that is expected to be implemented. The profiles do not restrict functionality, but generate requirements for a minimum set of functionality (see [9]).

The OPC Foundation also defines a set of information models that provide a basic set of functionality. The Data Access specification (see [10]) provides a basic information model for typical process control data. The Alarm and Condition specification (see [11]) defines a standard information model for Alarms and Conditions. The Programs specification (see [12]) defines a stand information model for extending the functionality available via method calls and state machines. The Historical Access specification (see [13]) defines the information model associated with Historical Data and Historical Events. The aggregates specification (See [14]) defines a series of standard aggregate functions that allow a client to request summary data. Examples of aggregates include averages, minimums, time in state, Standard deviation, etc.

The OPC UA standard also provides the concept of a view. Views allow a server to subset the functionality and information model that is exposed to a given user. They are typically server defined and can be used to provide a window into a process that is tailored to a specific class of users, such as operators or maintenance workers.

NAMESPACES

OPC UA allows information from many different sources to be combined into a single coherent address space. Namespaces are used to make this possible by eliminating naming and id conflicts between information from different sources. Namespaces in OPC UA have a globally unique string called a NamespaceUri and a locally unique integer called a NamespaceIndex. The NamespaceIndex is only unique within the context of a Session between an OPC UA Client and an OPC UA Server. All of the web services defined for OPC UA use the NamespaceIndex to specify the Namespace for qualified values.

There are two types of values in OPC UA that are qualified with Namespaces: NodeIds and QualifiedNames. NodeIds are globally unique identifiers for Nodes. This means the same Node with the same NodeId can appear in many Servers. This, in turn, means Clients can have built in knowledge of some Nodes. OPC UA Information Models generally define globally unique NodeIds for the TypeDefinitions defined by the Information Model.

QualifiedNames are non-localized names qualified with a Namespace. They are used for the BrowseNames of Nodes and allow the same Names to be used by different information models without conflict. The BrowseName is used to identify the children within a TypeDefinitions. Instances of a TypeDefinition are expected to have children with the same BrowseNames. TypeDefinitions are not allowed to have children with duplicate BrowseNames; however, Instances do not have that restriction.

COMPANION SPECIFICATIONS

An OPC UA companion specification for an industry specific vertical market describes an information model by defining ObjectTypes, VariableTypes, DataTypes and ReferenceTypes that represent the concepts used in the vertical market. Table 1 contains an example of an ObjectType definition.

Table 1 – Example *ObjectType* Definition

Attribute	**Value**				
BrowseName	WidgetType				
IsAbstract	True				
Reference	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the *BaseObjectType* from OPC UA Part 5[7].					
HasProperty	Variable	Color	String	PropertyType	Mandatory
HasProperty	Variable	Flavor	LocalizedText	PropertyType	Mandatory
HasProperty	Variable	Rank	Int32	PropertyType	Mandatory

The BrowseName is a non-localized name for an ObjectType.

IsAbstract is a flag indicating whether instances of the ObjectType can be created.

The bottom of the table lists the child nodes for the type. The Reference is the type of reference between the Object instance and the child Node. The NodeClass is the class of Node. The BrowseName is the non-localized name for the child. The DataType is the structure of the Value accessible via the Node (only used for Variable NodeClass Nodes) and the TypeDefinition is the ObjectType or VariableType for the child. The ModellingRule indicates whether a child is Mandatory or Optional Figure 7 visually depicts the ObjectType defined in Table 1 along with two instances of the ObjectType.

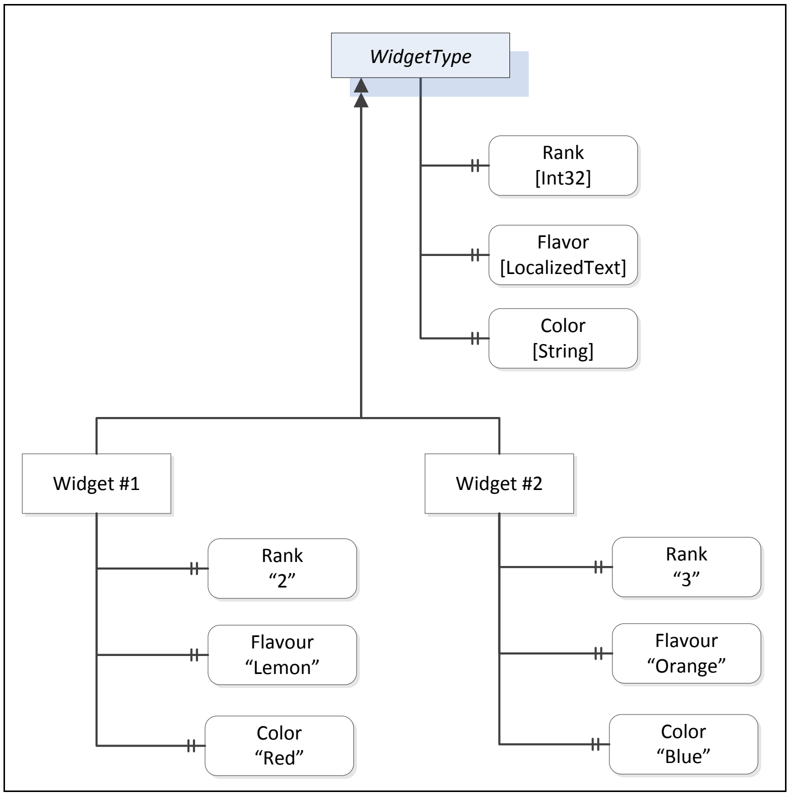


Figure 7 – A Visual Representation of the Sample ObjectType

ISA 95 INFORMATION MODELS:
ITS ROLE IN IMPROVING BUSINESS VALUE

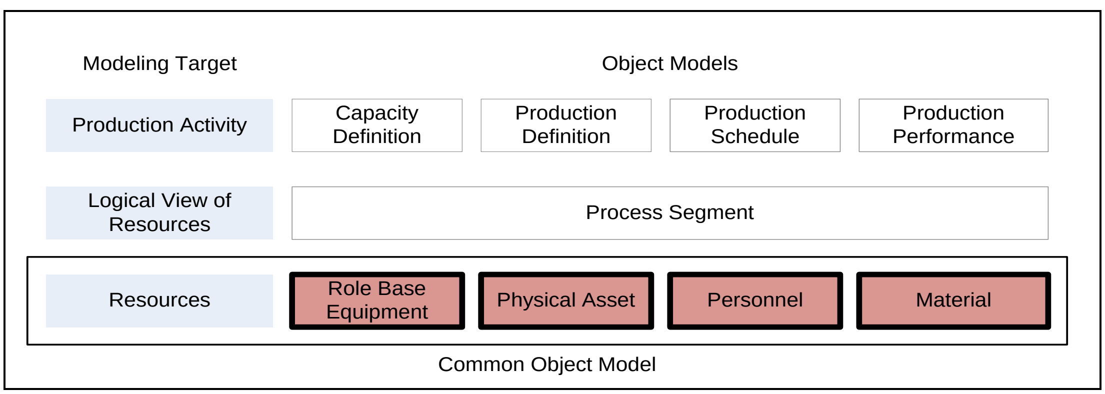
The ANSI/ISA 95 Enterprise/Control System Integration standard [1], also released as IEC 62264 [2], defines five levels of activities in a manufacturing organization. Generally, automation and control support Levels 1 and 2, MOM systems support Level 3, and business Enterprise Resource Planning (ERP) systems support Level 4 activities. The ISA 95 levels are shown in Figure 1.

- **Level 0** defines the actual physical processes.
- **Level 1** elements are the sensors and actuators attached to the control functions in automation systems.
- **Level 2** automation and control systems have real-time responses measured in sub-seconds and are typically implemented on Programmable Logic Controllers (PLC), Distributed Control Systems (DCS), and Open Control Systems (OCS). Level 3 typically operates on time frames of days, shifts, hours, minutes, and seconds. Level 3 functions also include maintenance functions, quality assurance and laboratory functions, and inventory movement functions, and are collectively called Manufacturing Operations Management.

- **Level 3** activities, including SCADA (Supervisory Control and Data Acquisition) systems for monitoring the process and providing operator control, batch control systems for execution of recipes, data historians for the collection and preservation of time based data from Level 2 systems, recipe and document management systems for managing recipes and workflow instructions, detailed scheduling, campaign management or work dispatching, and work or product tracking. Level 4 is called Business Planning and Logistics.
- **Level 4** typically operates on time frames of months, weeks, and days. Enterprise Resource Planning (ERP) logistics systems are used to automate Level 4 functions.

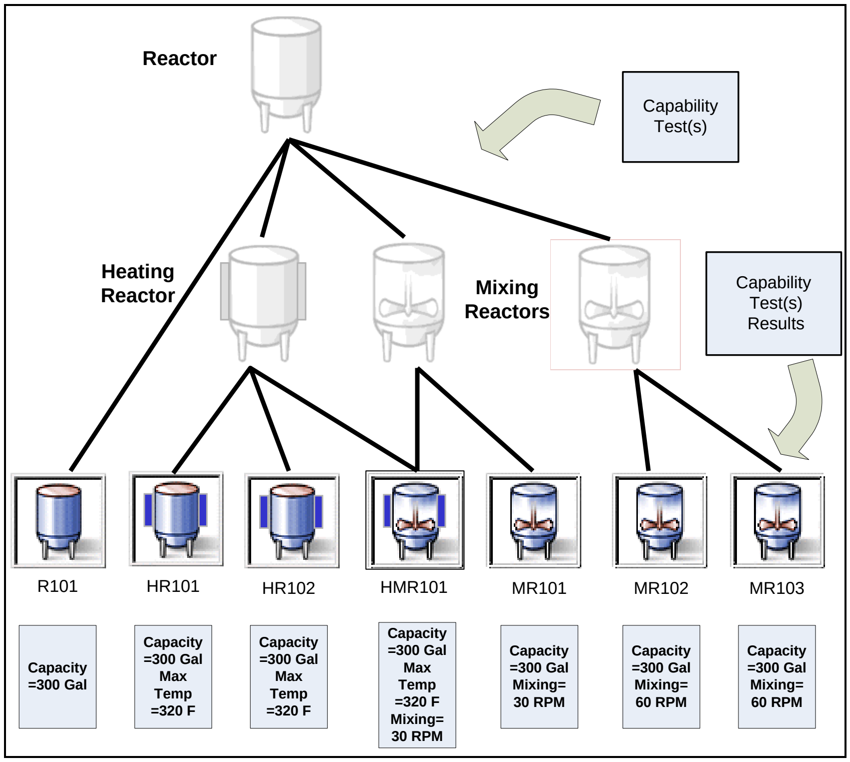
It is important to remember that each level has some form of control and each level has its own definition for real-time. Level 3 systems consider real-time to mean information is available a few seconds after shop floor events occur. Level 4 systems consider real-time to mean that logistics and material information is available daily or within a few hours after the end of a shift.

The OPC UA ISA-95 mapping is designed for information exchange across Level 3 systems or between Level 2 and Level 3 systems. Specifically this would involve information exchange between MES, WMS, LIMS, AM, PLC and DCS systems. This information exchange in real-time, often requiring transaction times measured in seconds or subsections, in order to allow workflows and recipes to execute in a timely manner. ISA 95 defines four primary types of information that often must be exchanged among MOM systems and between ERP systems and MOM systems, these types are; information about material and the properties of materials, information about equipment as it pertains to the operations being performed, information about the physical assets that make up the equipment, and information about personnel and their roles and qualifications.

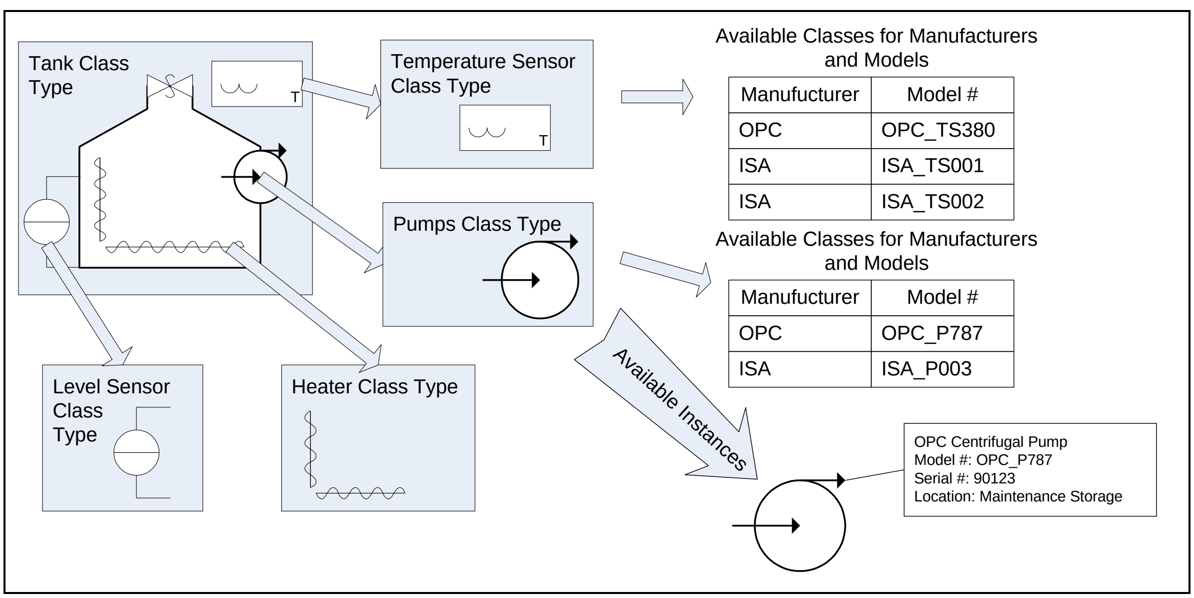


Material Class, Material Definition, Material Lot, Material Sub-lot, and Material Test Information – This is the definition of the lots, sub-lots, material definitions, and material classes involved in production. This information allows Level 3 and Level 4 systems to unambiguously identify material specified in production schedules and consumed or produced in actual production.

Equipment Class, Equipment, and Capability Test Information – This is the definition of the roles that equipment and equipment classes perform in production, inventory management, and quality test labs. This information may be used to associate production with specific equipment as part of a production record, or with equipment classes to schedule production and allocate costs.



Physical Assets, Physical Asset Classes, and Physical Asset Capability test information – This is an identification of the specific physical asset (by serial number or asset ID) used in manufacturing operations. It also includes the make and model information that identifies the type of physical asset and its properties.



Personnel Class, Person, and Qualification Test Information – This is the definition of the persons and personnel classes (roles) involved in production. This information may be used to associate production with specific persons as part of a production record, or with personnel classes to allocate production costs.

THE SOLUTION

The proposed solution is comprised of a standard OPC UA aggregation system, in which multiple data sources can provide data. These data sources could be input into a single OPC UA Server or they could be unique vendor provided OPC UA servers or even wrapped OPC Classic servers. The system would have Standard OPC UA based calculation engine to compute any required KPI or other aggregation data. The System would also include an aggregating server that would compose the multiple data sources into a single uniform information model that would include standard graphics, displays, reports and alarms. The choice of the final architecture would be up to the ultimate client, since it is easy to configure and to relate metadata from multiple systems. All of the communication would be occur via the highly secure and high performance OPC UA interface. Client access to data can be restricted by the use of standard OPC UA views. Thus allowing a fully integrate standard view of the system.

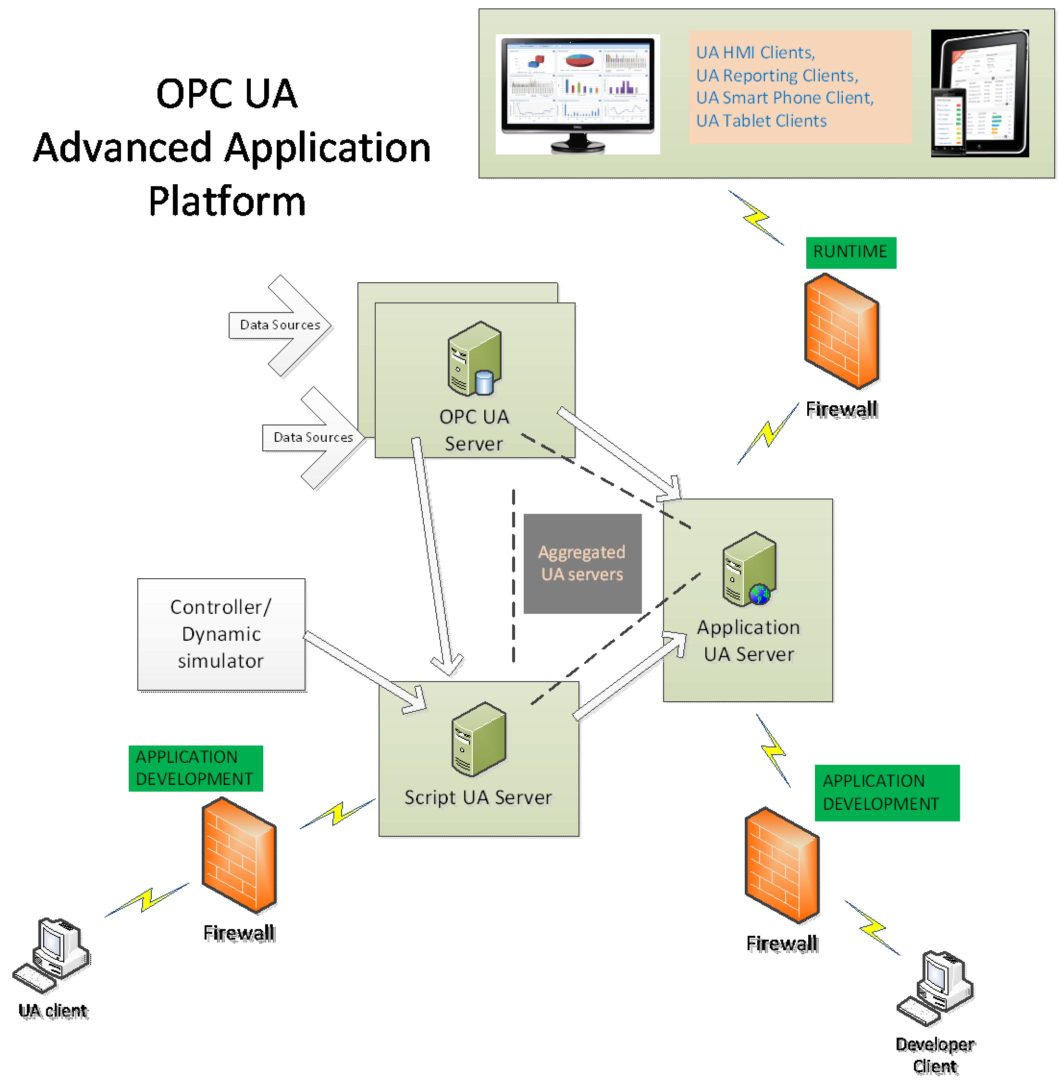


Figure 8 - Architecture

A system can be defined for a set of standard logical objects, using the OPC UA ISA-95 Equipment Model. These Equipment Objects can include standard KPI information models. The system can also be defined for a set of standard physical objects, using the OPC UA ISA-95 Physical Asset Model. These Physical Assets can include standard maintenance procedures.

As defined in ISA-95 the Physical Asset that is currently assigned to logical equipment can be replaced. Since the KPI's are part of the logical equipment model, if a different physical device is utilized, the KPI would still apply.

If Physical devices are replace with different vendors device then only the maintenance procedures for the new devices would be required. As long as the new physical Asset is capable of generating the information required for the logical deice KPI, the KPI can remain unchanged.

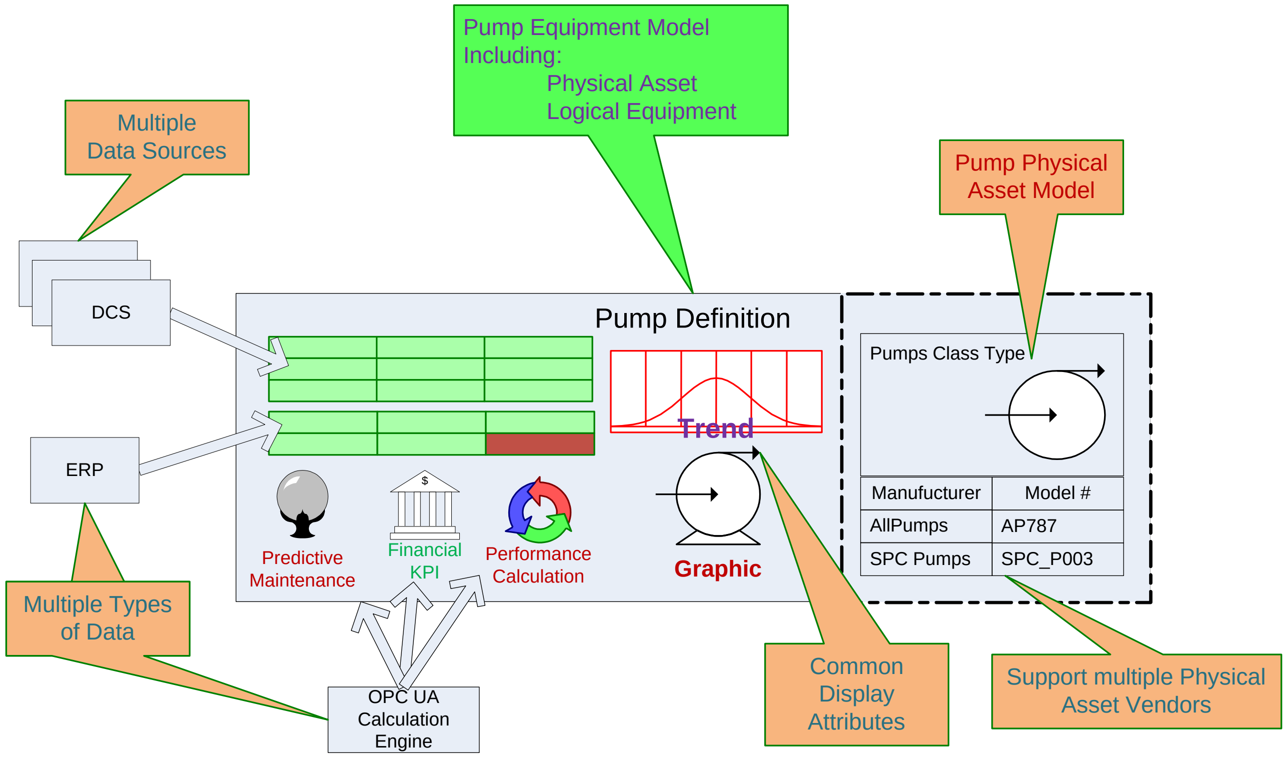


Figure 9 - The Generic Manufacturing Intellegence Object Model

HMI displays, reports or other reporting aspect of the system can be defined utilizing the concepts of browsePaths, which would allow the same display or report to be used for a number of different instances. The infrastructure for executing the KPI can also be shared between the instances.

When this system is mapped to a different plant, various aspects of the system can remain the same. Only the aspect that are unique to the given plant would need to be updated. Since the entire system is typed based, each plant can have any number of instances or a different instance structure.

The overall system is illustrated in Figure 10

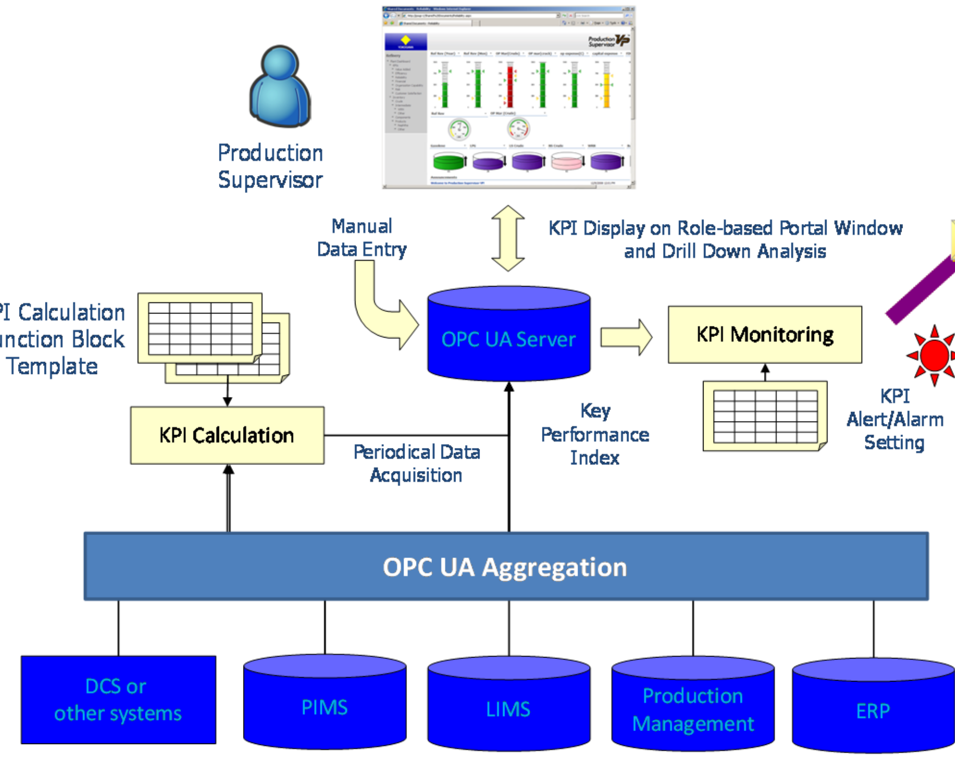


Figure 10 – Solution Overview

CONCLUSION

THE IMPORTANCE OF CONTEXT FOR BUSINESS AGILITY – VALUE ADDITION

Information is data with a context. If only raw data is provided to a user even if it is accurate it would not help in any way with making any decisions or understanding the current status. In today's environment where profitability is the key to survival of the business it is important to make sure decisions are made accurately and timely.

OPC UA provides an information modeling capability that allows industry specific models to be integrated with standards based models providing a robust information model that any client can access. This allows organizations to leverage the power of the information model specific to their organization. It facilitates visualizing the information as relevant KPIs, allowing the organization to make informed decisions improving the productivity and ensuring consistent operations.

Since OPC UA information models are based on configuration rather than implementation it easy to create a context based on user needs within a short time there by adding a great deal of value to users.

REFERENCES

[1] ANSI/ISA-95.00.01: Enterprise-Control System Integration - Part 1: Models and Terminology, 2010
[2] IEC 62264-1: Enterprise-control system integration – Part 1: Models and terminology, Edition 1.0, March 2003
[3] IEC 62541-1: OPC Unified Architecture - Part 1: Overview and Concepts
[4] IEC 62541-2: OPC Unified Architecture - Part 2: Security Model
[5] IEC 62541-3: OPC Unified Architecture - Part 3: Address Space Model
[6] IEC 62541-4: OPC Unified Architecture - Part 4: Services
[7] IEC 62541-5: OPC Unified Architecture - Part 5: Information Model
[8] OPC UA for ISA-95 Common Object Model Release
[9] IEC 62541-7: OPC Unified Architecture - Part 7: Profiles
[10] IEC 62541-8: OPC Unified Architecture - Part 8: Data Access
[11] IEC 62541-9: OPC Unified Architecture - Part 9: Alarms &Conditions
[12] OPC Unified Architecture - Part 10: Programs
[13] OPC Unified Architecture - Part 11: Historical Access
[14] OPC Unified Architecture - Part 13: Aggregates

AUTHORS

PAUL HUNKAR is president of DS Interoperability, independent consulting firm, specializing in design and development of software systems. He has been running DS Interoperability for 3 years, his client list includes Yokogawa, ABB, Unified Automation, RovoSys, the OPC Foundation…. Paul is member of the Technical Advisory Council of the OPC Foundation. He is editor for the Profiles, Security, and Alarm & Condition parts of the OPC UA Specification. He was the Director of Certification for the OPC Foundation and the chair of the OPC Foundation Compliance Working Group. He had previously worked for major Process Automation vendors for more than 25 years in a variety of engineering roles including designing, constructing and managing development of advanced control applications, operator interface systems, historical systems and investigating new technologies. Paul has a Bachelors Degree in Computer Engineering from the University of Michigan and a Masters Degree in Computer Engineering from Case Western Reserve University.

DSInteroperability,
4012 Deacon Ct, 44236 Hudson, Ohio,
Tel. +1 440 337 4161,
E-Mail: **Paul.Hunkar@DSInteroperability.com**

SUBRAMANIAN (SUBBU) PERIASWAMY is a principal systems architect with Yokogawa in the Indian development center. Subbu has worked on products related Manufacturing operations management (MOM) and Advanced operation assistance. Subbu contributed as an architect for Real time production organizer(RPO) which is a suite of platform applications developed specifically to bridge the gap between production planning and process control, providing a fully integrated business to process (B2P) production environment. Subbu has also worked on products related to Procedure automation and is keen in following ISA – 106.

Yokogawa IA Technologies India Pvt Ltd
06th Floor Umiya Business bay, Cessna Business park
ORR, Kadubeesnahalli, Marathalli Post, Bangalore -560037
Tel. +91-80-67925130
E-Mail: Periaswamy.subramanian@yti.yokogawa.com