Machine-To-Supervisory Communication Framework based on OPC Unified Architecture

Pedro Reboredo - Research and Development,
Bosch Rexroth AG

Presented at the ODVA
2012 ODVA Industry Conference & 15th Annual Meeting
October 16-18, 2012
Stone Mountain, Georgia, USA

Abstract:

The requirements on machine integration grew continuously over the last decades. Due to the variety of fieldbus standards and the complexity of a heterogeneous field device landscape this resulted in high implementation efforts for the application. This paper describes a conceptual framework for machine-to-supervisory communication done via protocol independent OPC UA objects integrating sercos and CIP based field device networks. It is based on an OPC UA server-client-network architecture via standard TCP/IP protocols. This approach allows interoperability between machine and supervisory systems and reduces the implementation time and costs.

Keywords:

CIP, sercos, OPC UA, Information Modeling, Factory Communication, Supervisory-To-Machine Communication

1. Introduction

Industrial manufacturing is still an important factor for the global economy. Automated production machines enable a constant product quality and ensure wealth of industrial nations. The variety of product configurations and the decreasing product life cycle times requires flexible manufacturing systems with a holistic approach for machine optimization. This ensures business investment turnover and enables transparent productivity awareness. A central requirement therefore is a universal communication framework that ensures overall interoperability.

2. Machine Optimization Requirements

Holistic machine optimization requires control feedback loops between supervisory and machine level. This means that a supervisory system has a direct access to actual values of the technical process. In fact due to the variety of field device suppliers this implies the implementation of a variety of communication protocols for supervisory systems. Intermediate gateways are needed to gain access into machine networks.

2.1. Communication Architecture

According to ISA-95 a factory environment can be shown as a hierarchy wherein communication instances are categorized into different levels. In analogy to the well-known automation pyramid these levels describe the increasing aggregation level and decreasing cycle time from shop floor to the top floor, from ISA-level 0 to ISA-level 4 [1]. Figure 1 shows these levels with their associated explanations.

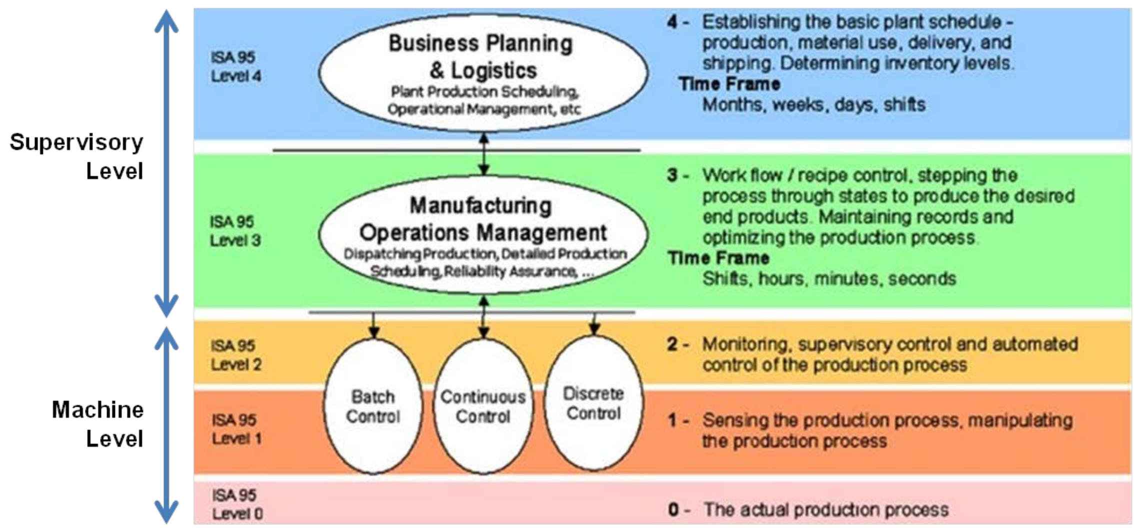

Figure 1: ISA-95 Manufacturing Levels according to [1] from [2]

Supervisory level systems include systems such as:

- Enterprise Resource Planning (ERP)
- Manufacturing Execution Systems (MES)
- Supervisory Control and Data Acquisition (SCADA)
- Human Machine Interfaces (HMI)

Machine level systems include systems and components such as:

- Controllers
  (Numeric Controllers, Motion Controllers, Programmable Logic Controllers, Robot Controllers)
- Servo Drives
- Digital IO's
- Analog IO's

Typically a machine consists of one or more machine controllers in connection to other field devices. These devices are in direct influence to the technical (production) process.

2.2. Requirements for Controller-To-Field level Communication

On machine level different fieldbus standards have been established due to the fact that each has its own history and advantages. During the last years Ethernet-based fieldbus standards have achieved acceptance for industrial applications. One of the advantages is the availability of standard hardware components, since Ethernet is well known from the office world [3]. For usage within machine level communication additional requirements have to be fulfilled since the production differs from the office environment.

Real Time Data Transmission

The communication e.g. from a numeric controller to a servo drive within a machine tool has hard requirements regarding the transmission of data between these two devices. The sent data has to reach the receiver within a defined time slot. Synchronization plays a major role in machine accuracy. Sophisticated real time Ethernet based fieldbus standards (RTE), e.g. sercos III with its modified Ethernet data link layer to achieve or highly synchronous data delivery or EtherNet/IP using unmodified Ethernet in conjunction with IEEE 1588 precision time synchronization to achieve CIP Motion, can be combined with standard Ethernet to allow integration of commercially-off-the-shelf (COTS) devices within the rest of the fieldbus network.

Availability of capable components

One other key requirement is the availability of products and components. There’s a need for both real time and non real time capable communication components. This includes hardware (ICs, Controllers) as well as software (communication stacks, software libraries) components.

2.3. Requirements for Supervisory-To-Machine Communication

IT-based supervisory systems mostly integrate gateways to communicate to the machine level. These gateways are mostly developed using specific protocols implementing dedicated application layers on client and server. To minimize engineering effort during implementation and operation some requirements have to be fulfilled.

Open Standards:

One of the most important requirements is the usage of open standards for the communication between supervisory systems and machine level devices. This reduces vendor dependencies and therefore minimizes the risk for manufacturing companies. Furthermore it allows a market competition for systems and software suppliers and enabling communication through different application layer protocols over the same network and subnets. During the last years the usage of layer-3 IP (Internet Protocol)-based communication mechanisms has become established.

Semantic Interoperability:

Interoperability enables the communication between devices of different suppliers. It can be differentiated between syntactic and semantic interoperability. Semantic interoperability ensures the correct interpretation of data by means of content compliance. Syntactic interoperability enables the transmission of data contents. It is fundamental for semantic interoperability since it defines data formats. While fundamental IP-based communication doesn’t provide semantic interoperability by itself.

Security Mechanisms:

Security mechanisms prevent machine data or interfaces from being misused. Operating machines within a factory network requires appropriate mechanisms. These can be e.g. user or group rights administration, user authentication or data encryption

Communication Abstraction:

Modern concepts of IT-based software systems e.g. ERP systems use appropriate abstraction mechanisms for communicating to a factory network due to the known variety of protocols. Middleware-based infrastructures can help abstracting the underlying communication mechanisms.

3. Information Modeling:

Semantic interoperability is achieved by defining a common information interpretation. As a central fundament within this framework communication objects (entities) have to be defined including their semantic building relationships. These entities are defined in an information model and can be accessed through defined mechanisms. For machine-to-supervisory communication this concept hides communication mechanisms of the machine level. Using the application layer of the established protocols CIP and sercos will maximize the use case coverage. Therefore these data model concepts are analyzed.

3.1. CIP Data Model [6]

CIP data are represented as attributes, organized within Objects. A CIP node (an Ethernet/IP communicating device) includes a collection of Object instances. Objects share different Attributes and Services. Object classes group several Objects into logical units.

A data value can be addressed through *Device.Class.Instance.Attribute.* 8Bit, 16Bit and 32Bit addressing is supported. Typically 8Bit addressing is used for attributes. Figure 2 shows an example attribute addressing.

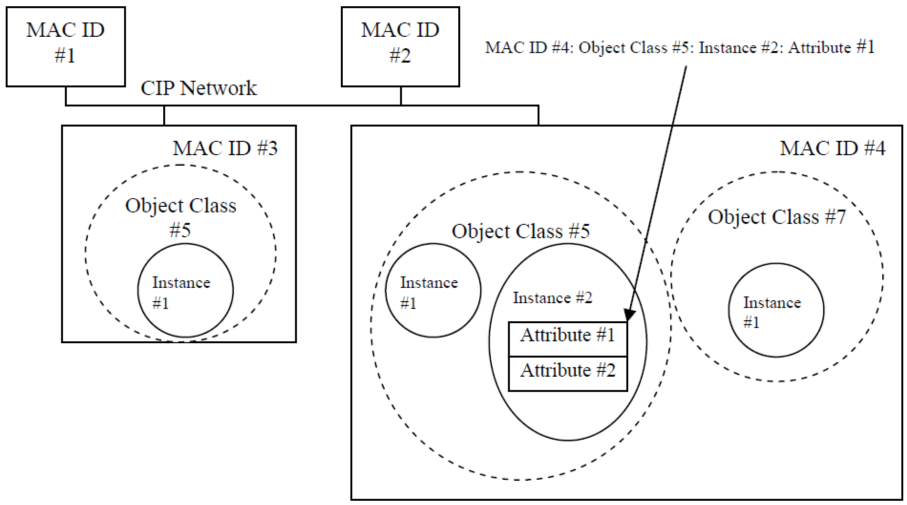

Figure 2 : Addressing a CIP Attribute from [6]

CIP differentiates between Communication and Application Objects. Communication Objects are used to manage the transport of information between nodes. Application Objects implement product-specific features.

3.2. sercos Data Model [4], [5]

sercos Data Profiles
The sercos data model is organized in to three profile classes:

- *SCP - sercos Communication Profile*
  To communicate within a sercos network each device implements SCP (sercos communication profile). This set of parameters is related to the communication configuration including cycle time.

- *GDP - Generic Device Profile*
  This profile includes device specific parameter sets that are independent from a device category including e.g. diagnosis, archiving, administration or identification

- *FSP - Function Specific Profile*
  Function specific profiles include parameter sets for dedicated device class functionality. It supports interface to different device resources including e.g. drive control parameters.

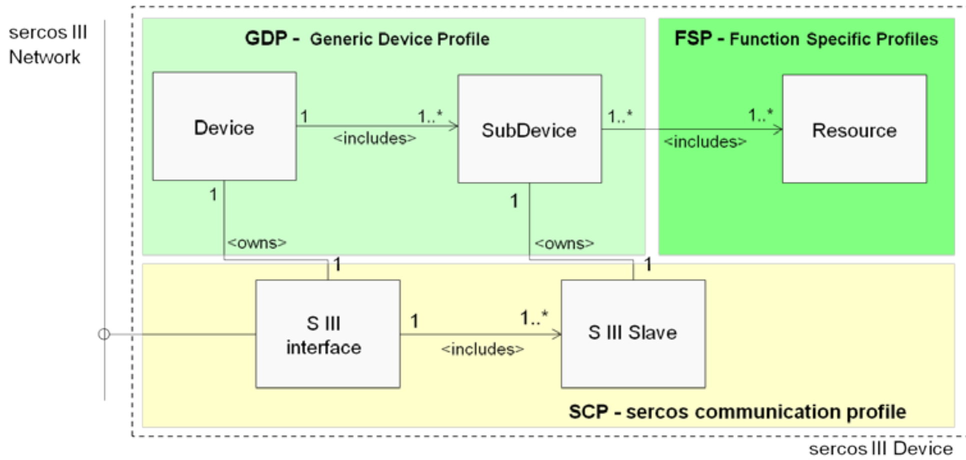

Figure 3: Overview of sercos Data Profiles according to [5]

Each device includes several profiles covering different functional areas. This leads to modularity within a device regarding the aspect of product configuration. Figure 3 gives a schematic overview of these profile classes including entities and dependencies:

sercos Parameter Model:
Each sercos parameter is identified using a 4 byte identification number (SIII IDN). Figure 4 shows its composition:

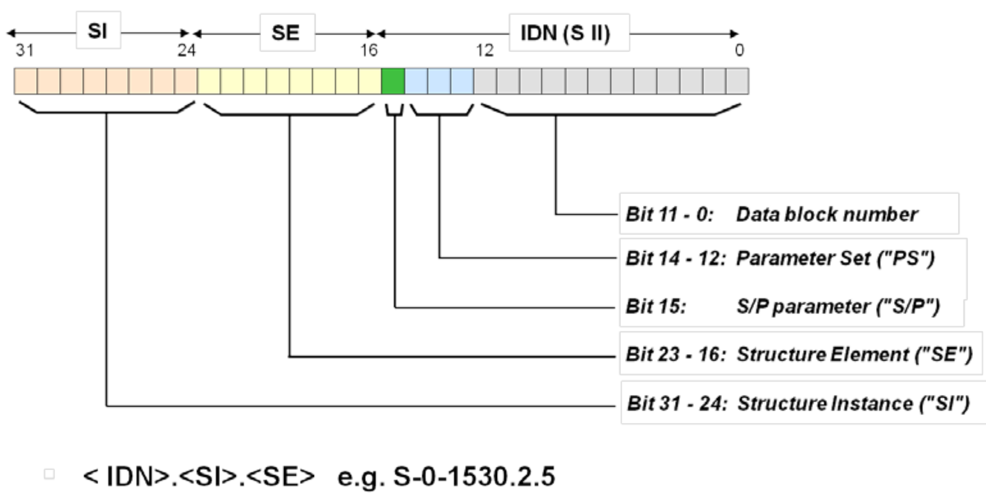

Figure 4 : sercos IDN structure according to [5]

IDN SII (sercos 2 IDN) includes the *Data Block Number*, *Parameter Set* and *S/P Parameter* field. The Data Block Number defines the block in which the data is included. There can be more than one set of one data block which is defined with the Parameter Set Index (as Bit 14-12). The addressed block can be a Standard (S) or Product Specific (P) parameter. Product specific parameters can be defined by a product manufacturer. Since sercos 3 it's possible to define parameter groups. These are represented by a *Structure Element* and a *Structure Instance* bitfield. A structure consists of different elements (SE) and can be instantiated (SI) at least once. This leads to a well-ordered data access mechanism.

Each parameter consists of different attributes. Figure 5 lists these attributes.

element No.	Description	Requirement
1	IDN	mandatory
2	Name	optional
3	Attribute	mandatory
4	Unit	optional
5	Minimum input value	optional
6	Maximum input value	optional
7	Operation data	mandatory
NOTE Elements 5 and 6 are mandatory for cycle time parameters (S-0-1050.x.10, S-0-1002).		

Figure 5 : sercos parameter attributes from [5]

IDN, Attribute and Operation Data are mandatory. IDN includes the Identification Number. Attribute is a bitfield containing different information e.g. data type. The Operation Data contains the value itself. Typically it's addressed during cyclic communication.

4. OPC Unified Architecture [7]

OPC Unified Architecture (OPC UA) is a middleware-technology that extends the OLE for process control (OPC) standard. OPC has a huge number of installed systems from different vendors. It's a well established technology for data exchange within process and factory automation, building automation and other applications. OPC UA as an evolution of OPC integrates functionality of former different OPC servers (e.g. Data Access, Alarms and Events, Historical Access) within one single server. It consists of a SOA based and platform independent technology. A coexistence of OPC and OPC UA is also possible. The communication between OPC UA clients and servers is Ethernet-based and supports TCP/IP with binary or XML content.

Data Modeling [8]

OPC UA provides the possibilities of semantic modeling with an integrated information model. The included information is organized with nodes. A node includes different attributes and references to other nodes. Objects are derived from nodes including Variables and Methods (see Figure 6). Variables can be simple (e.g. single value) or complex (e.g. data structure).

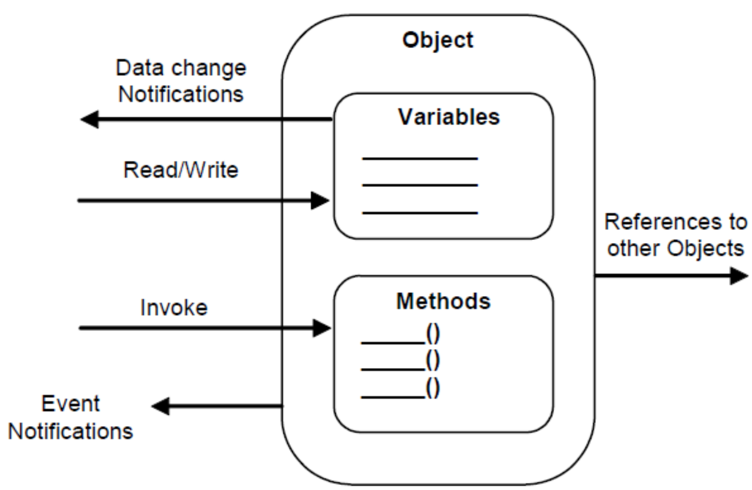

Figure 6: OPC UA Object model from [9]

OPC UA Services [10]
The model structure is presented within the server address space and can be accessed through different predefined and standardized services. These service sets include e.g.:

- *Discovery*
  This service set includes mechanisms for server and endpoint discovery. Servers can register themselves to one known discovery server and provide own discovery services for connection establishing. A separate discovery server can be useful for multi-server and multi-client architectures, specially within complex factory networks.

- *Secure Channel*
  Security mechanisms are summarized within this service set. A secure channel can be established between Client and Server supporting both signed and encrypted messages. The security mechanisms are integrated within the Communication Stack.

- *Methods*
  This service set supports function calls of methods related to Objects. This includes the interface definition (input and output arguments) by method properties. Furthermore browse and query services for method discovery are available.

5. Framework concept

This framework's architecture consists of an OPC UA based communication between supervisory systems and machine level. OPC UA is chosen because the requirements can be fulfilled with this technology. Figure 7 describes the framework design of this server-client network architecture. The supervisory system has access to the machine system via an OPC UA interface. This can supplementary be abstracted with a communication abstraction layer.

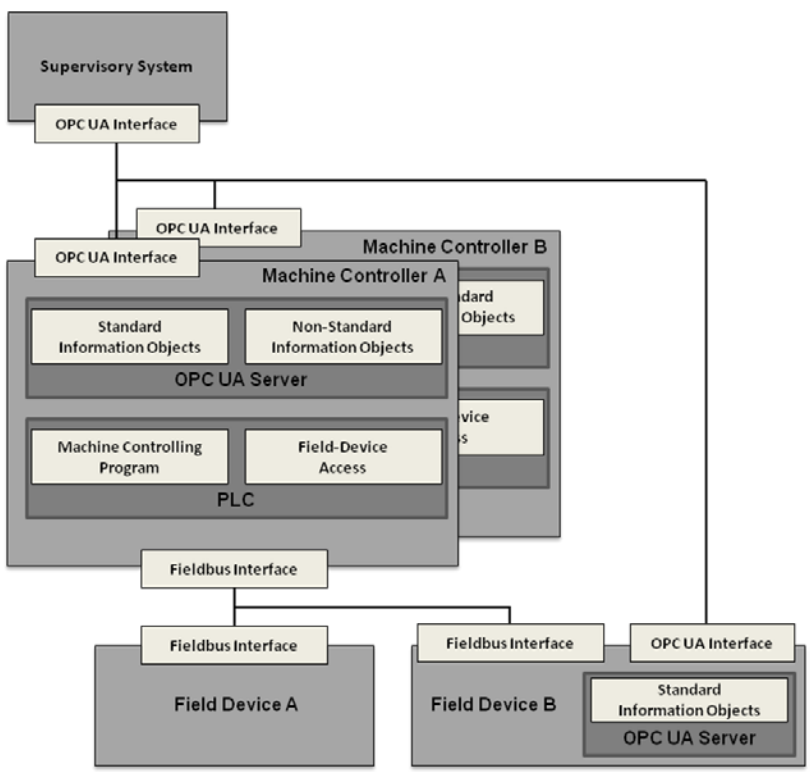

Figure 7 : Overview of the Communication Framework

Machine Information Model

The machine information model within the OPC UA Server (Figure 7) consists of standardized and non-standardized information model parts. The interoperability between different device vendors will be achieved with standardized models. Both are OPC UA data modeling based. Standard models can include logical machine data groupings, e.g. Energy Management, Remote Condition Monitoring [11]. Meanwhile the non-standard part enables flexibility for further machine data e.g. application specific data required by end users. The internal transmission of data from PLC to its embedded OPC UA server is controller-dependent. Since OPC UA is an established technology a number of SDKs are available to help the controller manufacturer embed a server.

Field Device Communication

The communication channel between a machine controller and the field devices can be realized by a fieldbus. In this case the access from the supervisory system to the field device can be realized by PLC-enabled mechanisms on controller level. That data can be exposed through an embedded OPC UA server in the controller with non-standard Information models. Alternatively the field devices can implement an OPC UA server to provide a supplementary channel for a direct communication between supervisory systems and field devices.

6. Conclusions and Outlook

Machine-to-supervisory communication is essential for holistic optimization of machinery. This paper shows a suitable communication framework to solve the problems of integration. Standardized information models facilitate the implementation efforts for machine data integration. CIP, sercos and OPC UA data modeling concepts are introduced. Still the standardization of information models has to be advanced. Actual activities within the ODVA O.M.I. task force will address this approach to ensure optimized interoperable machine-to-supervisory communication possibilities.

References:

[1] International Society of Automation. *ANSI/ISA-95.00.01-2010 (IEC 62264-1 Mod)*

[2] Bolton, Ray and Tyler, Stephen. *PQLI Engineering Controls and Automation Strategy*. In *Journal of Pharmaceutical Innovation 2008, Volume 3, Number 2, Pages 88 to 94*

[3] Klasen, Frithjof; Oestreich, Volker and Volz Michael (Publisher). *Industrielle Kommunikation mit Feldbus und Ethernet, 2010*

[4] sercos International. *General Overview and Architecture. 16th December 2011*

[5] sercos International. *Generic Device Profile. 16th December 2011*

[6] ODVA, Inc. *The CIP Networks Library Volume 1: Common Industrial Protocol. November 2011*

[7] Lange, Jürgen; Iwanitz, Frank and Burke, Thomas J. *OPC von Data Access bis Unified Architecture. VDE Verlag GmbH, 2010*

[8] Mahnke, Wolfgang; Leitner, Stefan-Helmut and Damm, Matthias. *OPC Unified Architecture. Springer 2009*

[9] OPC Foundation. *OPC Unified Architecture Specification, Part 3: Address Space Model*, Release 1.01, February 6, 2009

[10] OPC Foundation. *OPC Unified Architecture Specification, Part 4: Services*, Release 1.01, February 6, 2009

[11] ODVA, Inc. *Industry Brief: Update on ODVA Machinery Initiative in cooperation with OPC Foundation and Sercos International*. April 2012