EDA system for manufacturing industries in IIoT environment

[1*]Yunhee Kang

[1]*Division of Information and Communication, Baekseok University, Cheonan, Korea*
[*]*yhkang@bu.ac.kr*

Abstract

Manufacturing process has been changed to more complex in many industry areas. To collect and use data from the equipment in the factories is required to support the greedy demand for data exposed by the process analysis and control applications. IIoT focuses on how smart machines, networked sensors and sensor analytics can help improve business-to-business (B2B) initiatives across a wide variety of manufacturing industries. In this paper, we design a software architecture based on SOA for EDA system of semiconductor manufacturing filed in IIoT environment.

Keywords*: EDA, IoT, IIoT, SOA, manufacturing industries*

1. Introduction

Information distributes through the Internet are generated as the process of the data produced by human. However, the Internet environment is advancing so that the producer and the consumer of information made go together with all the objects with the connection of physical objects to the Internet by taking charge of sensing and actuation function, data processing and communication function. Internet of Things (IoT) is defined as the sharing of information through the connection of objects to the Internet, and the IoT environment is rapidly being established with the supply of various sensors [1-2].

In the world of manufacturing, the version of IoT, the Industrial Internet of Things (IIoT), is a logical extension of automation and connectivity that has been a part of the plant environment. Especially today, manufacturing is becoming highly-automated and IT-driven or simply put, smart. IIoT is based on primarily in the area known as machine-to-machine (M2M) communication. Manufacturing process has been changed to more complex in many industry areas. For instance semiconductor manufacturing is a highly complex production process composed of hundreds of steps. An interesting aspect of semiconductor manufacturing is that the front-end contributes 90% of capital costs and 80% of lead times. This result shows the fact that investments in equipment and processes have the highest share of manufacturing costs and any improvement in equipment utilization. It is required for manufacturers to continuously improve operational performance [3-4]. Hence to collect and use data from the equipment in the factories is required to support the greedy demand for data exposed by the process analysis and control applications.

Recently, many intelligent applications of semiconductor factories require variety and complex data types from integrated information framework. EDA can be used to get significantly higher trace data collection throughput, and the robust tool model in EDA provided better access to sensors and other key equipment variables useful for operational data monitoring.

This paper introduces a design of software architecture based on SOA for EDA system of semiconductor manufacturing filed in IIoT environment. As the composition of a single architecture for supporting the requirements of use-case of users of various areas is not adequate by the nature of IIoT environment.

[*] Corresponding Author
Received: Aug. 30, 2016, Revised: Sep. 14, 2016, Accepted: Sep. 26, 2016

The rest of paper is organized as follows: Section 2 depicts related works including IoT, IIoT and Web Services. In this section we describes the definition of those techniques and characteristics. Section 3 describes EDA architecture design for handling data from sensors and instrucments. Lastly, section 4 depicts the conclusion.

2. Related Works

2.1 The characteristics of IoT

The IoT is the connection – via the internet – of objects from the physical world that are equipped with sensors, actuators and communication technology [1-2]. This technology is looked at by a large variety of domains, such as manufacturing, healthcare and energy, to facilitate the development of new applications and the improvement of existing applications. To also enable the commercial exploitation of these applications, new types of business models must be developed. The high level of interest and hype surrounding the IoT is driven by the proliferation of everyday objects with an Internet connection. The IoT refers to the networking of physical objects through the use of embedded sensors, actuators, and other devices that can collect or transmit information about the objects.

2.2 IIoT and its application

Industry 4.0, Industrie 4.0 or the fourth industrial revolution is a collective term embracing a number of contemporary automation, data exchange and manufacturing technologies [3]. It facilitates the vision and execution of a Smart Factory. Within the modular structured Smart Factories of Industry 4.0, cyber-physical systems monitor physical processes, create a virtual copy of the physical world and make decentralized decisions. Over IoT, cyber-physical systems communicate and cooperate with each other and with humans in real time, and via the Internet of Services, both internal and cross-organizational services are offered and utilized by participants of the value chain.

The IoT is changing business models, increasing output, and automating processes across a number of industries. In the world of manufacturing, our own version of IoT, the Industrial Internet of Things (IIoT), is a logical extension of automation and connectivity that has been a part of the plant environment for decades, primarily in the area known as machine-to-machine (M2M) communication [4]. The IIoT is the part of the IoT that focuses on how smart machines, networked sensors and sensor analytics can help improve business-to-business (B2B) initiatives across a wide variety of industries, especially manufacturing. The IIoT movement is, of course, growing and expanding at least as fast as the IoT in the outside world because smart devices and connected sensors are proliferating in the plant as well. It also gives a great impact on the manufacturing field.

The main features of IIoT are as follows:

- horizontal integration through value networks to facilitate inter-corporation collaboration,
- vertical integration of hierarchical subsystems inside a factory to create flexible and reconfigurable manufacturing system
- end-to-end engineering integration across the entire value chain to support product customization

In this paper we mainly focused on vertical integration. A factory runs several physical and information subsystems, such as actuator and sensor, control, production management, manufacturing, and corporate planning. It is essential to vertical integration of actuator and sensor signals across different levels right up to the enterprise resource planning(ERP) level to enable a flexible and reconfigurable manufacturing system. By this integration, the smart machines form a self-organized system that can be dynamically reconfigured to adapt to different product types; and the massive information is collected and processed to make the production process transparent.

In manufacturing specifically, IIoT holds great potential for quality control, sustainable and green practices, supply chain traceability and overall supply chain efficiency. Semiconductor companies that want to capture the IoT's enormous growth potential might be tempted to move ahead quickly, without changing their existing operating model, but this could be a mistake. Manufacturers including semiconductor manufacturing have invested massively in IoT devices. In the industrial IoT environment, those devices are used for connecting physical things to many information systems including SCM, ERP, etc. There are some hurdles to handle the IoT manufacturing:

- Replacing existing hardware without process disruption
- Concern about service lock-in and flexibility
- Adhering to standards

2.3 Web Services

Web Services can solve the issue on how to have access to many kinds of applications from a variety of platforms due to XML-based Web Services that depend on the Internet. By W3C, a Web Service is defined to be the following: a Web Service is a software application identified by a URI, whose interfaces and binding are capable of being defined, described and discovered by XML artifacts and supports direct interactions with other software applications using XML based messages via HTTP like internet-based protocol [8].

The emergence of service-oriented architecture (SOA) as an approach for integrating applications that expose services presents many new challenges to organizations resulting in significant risks to their business. Particularly important among those risks are failures to effectively address quality attribute requirements such as performance, availability, security, and modifiability. The goal of Web Services is to provide high interoperability in distributed systems. In a typical web services scenario, a business application sends a request to a service at a given URL using the Simple Object Access Protocol (SOAP) protocol over HTTP. SOAP is a lightweight protocol designed for the exchange of information. Focused on distributed, decentralized environments, it provides a framework to invoke services across the Internet. The service, described in WSDL, receives the request, processes it, and returns a response. SOAP provides an envelope that encapsulates XML data for transfer through the web infrastructure [5-7]. SOAP provides a way of encapsulating calls to remote objects in XML format. You can see the name of the remote operation to be executed as well as the data being passed in and out of that remote operation.

Web services architecture: the service provider sends a WSDL file to UDDI. The service requester contacts UDDI to find out who is the provider for the data it needs, and then it contacts the service provider using the SOAP protocol. The service provider validates the service request and sends structured data in an XML file, using the SOAP protocol. Figure 1 shows the overall structure of Web Services architecture.

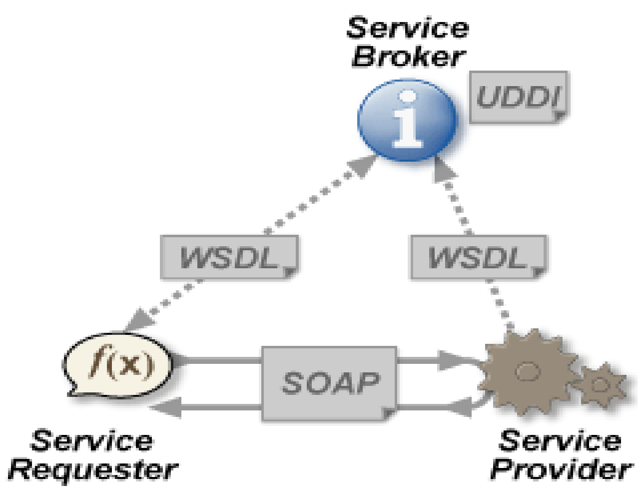

Figure 1. The architecture of Web Services

3. Design of EDA architecture

3.1 System requirement

A semiconductor manufacturing enterprise when considered at the level of the business process is, of course, very similar to other business processes. However, a look at the requirements for process automation at the level of the semiconductor manufacturing facility reveals that notwithstanding a number of commonalities between automating business processes and semiconductor manufacturing processes, several interesting issues appear in semiconductor manufacturing that do not arise in business processes. The computer integrated manufacturing (CIM) system of a semiconductor manufacturing facility needs to provide facilities for factory management, planning of factory operation, scheduling of factory resources, specification and manipulation of processes, monitoring overall factory performance, and monitoring and control of the equipment in the facility.

The system requirement for EDA is as follows:

- Host-independent data collection: clients setup and collect data near-real-time independent of host on/off-line status
- Security: only factory-authorized applications permitted to collect data, single point of control is enforced at factory level
- Self-describing interface: equipment structure, implemented state models, available data items & types, events, exceptions, and alarms can be learned at runtime from the tool
- Process control data: improvements in the ability to collect sampled data for up to 50 parameters per process chamber at a frequency 1% of shortest recipe step (worst case assumption is 1000 scalar parameters at 10Hz)
- Equipment operational data: visibility into module-subsystem-and actuator-level activity for facilitating equipment health monitoring, OEE, and diagnostics/troubleshooting

Basically EDA is a kind of event processing; event processing is a method of tracking, processing and analyzing streams of data represented status of equipment and value gathered from one about things that happen events. Complex event processing (CEP) is event processing that combines data from multiple sources [11] to infer events or patterns that suggest more complicated circumstances. The goal of complex event processing is to identify meaningful events (such as opportunities or threats) and respond to them as quickly as possible.

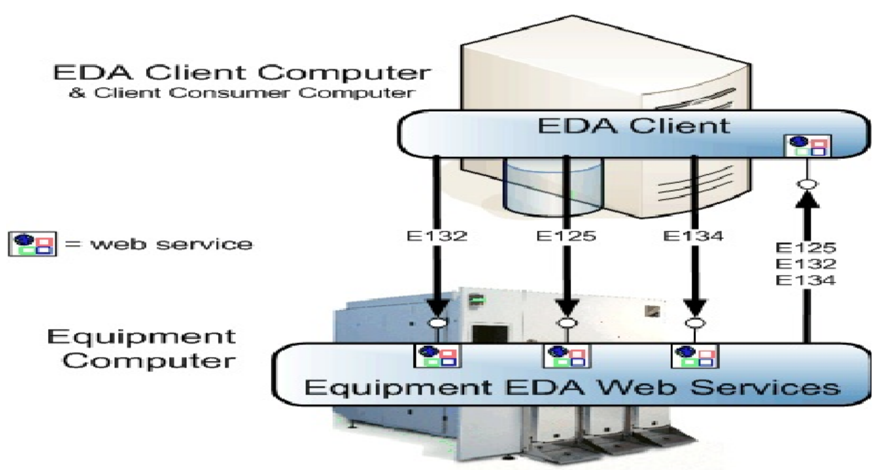

Figure 2. Overview of EDA standards

EDA can be used for how to detect symptom caused by errors happened from equipment. Managing the data proliferation to extrapolate intelligent conclusions in a timely manner will be critical in realizing the anticipated benefits of greater data accessibility enabled by Interface A. SEMI Interface-A provides a suite of specifications [14-16] for communication between data sources on the equipment and data consumers. In a factory control environment, data sources or servers are devices that compile and report process control data in formatted Extensible Markup Language (XML) reports called Data Collection Reports (DCRs). These reports are generated in response to a query sent to the device from a client, such as a data storage system or a supervisory plant floor controller. In SEMI

EDA terminology this query is called a Data Collection Plan (DCP) since it also carries information about the format, frequency and type of data to be sent out in the DCR. The specification allows multiple simultaneous connections to be maintained between clients and servers, employing the SOAP (Simple Object Access Protocol) messages over HTTP or HTTPS connections. Figure 2 shows the overall EDA standards.

3.2 Composition of IoT Sensing Information Collection System

System where the interaction between the objects and human are conducted like IIoT environment, and the system for the collection data collected based on the streaming required designing to support various services. The streaming-based data processing is operated through the interconnection with various services and contents, and the devices and objects of users exchanges information through messaging service for the application execution composed of services.

Messaging system provides two types of messaging models of publish-and-subscribe (pub/sub) and point-to-point queuing (P2P). Pub/sub model adequate for the realization of pipe-filter is utilized in the distribution of messages in the many-for-one type, and purpose of P2P model is for the one-on-one delivery of message [10-11]. The selection of these messaging models is determined in accordance with the messaging requirements of application. Especially, if the messages have to be distributed to various receivers, pub/sub model is adequate for messaging system. For component converting the data at pipe-filter pattern to deliver this to the remote component, make it utilize pub/sub-based middleware [12]. In pub/sub model, the message producer may transmit a single message to various consumers through virtual channel called Topic. The message producer operates without dependence on the consumer receiving the message. As this enables collection of occurred messages where the network connection is disconnected, it provides strength of collecting information in discontinuous network environment.

The collector collects data from numbers of agents and stores the data by adding meta-data. The collector processes data by regularly conducting filtering on the collected data. For this work, the filter uses Apache Flume or Storm conducting the role of message bus. The Thing internally possesses sensors and delivers the acquired information to the collector through the wire-wireless network without saving the acquired information at the storage device. At this moment, the application operating at the cloud computing resource processes acquired information and alarm information, and the information required for the application are delivered through the filter. Between the collector and the application, producer-consumer relationship is made through the filter, and provides high expandability through loose-coupled connection. In addition, abnormal status and error information are delivered through the interpretation of subscribed message in application.

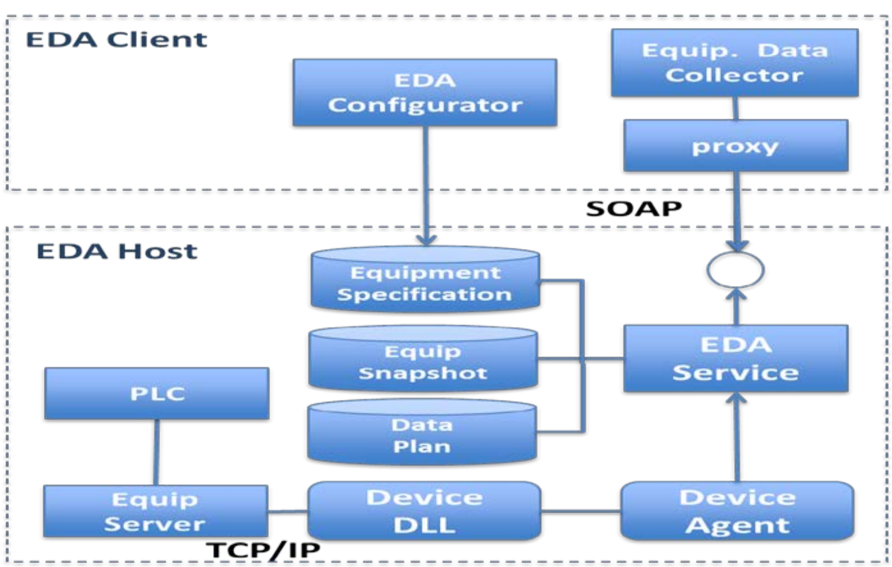

Figure 3. The EDA architecture

3.3 Experimental result

To study the performance of factory scale EDA in Ethernet interface, we design the EDA system in conjunction with a testbed composed of PLC associated devices, EDA Host and EDA client. Figure 3 shows the overall architecture for building an EDA application of the equipment in a semiconductor process.

In E125, metadata interface is well modular written in WSDL. The EquipmentMetadataManager provides 9 methods including GetTypeDefinitions, GetEquipmentStructure, GetEquipmentNodeDescriptions, etc. To get the overall structure of a equipment, an EDA client call a method GetEquipmentStructure in EquipmentMetadataManager. The response of this call GetEquipmentStructureResponseMessage provides a structure of a specific equipment predefined as CEM(Common Equipment Model) in E120 within a SOAP message. Figure 4 shows the SOAP message for GetEquipmentStructure method with input and output messages.

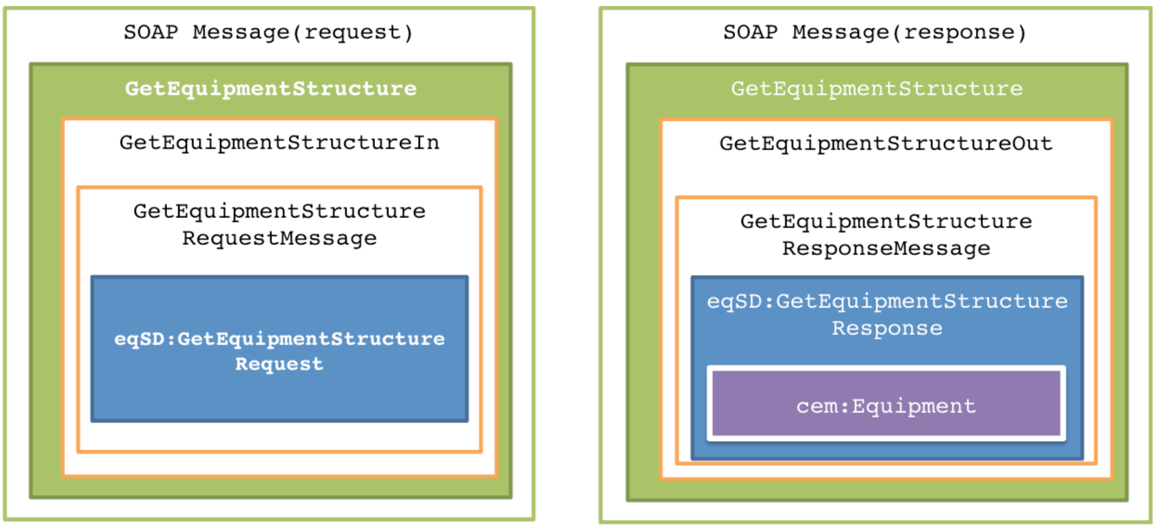

Figure 4. The SOAP message structure for GetEquipmentStructure method

The return value of GetEquipmentStructure method is represented as GetEquipmentStructureResponsemessage. It has two types of values: the equipment structure named cem:Equipment and the error message named ccs:Error in a XML schema as follows:

```
<xsd:complexType name="GetEquipmentStructureResponseType" final="#all">
  <xsd:choice minOccurs="0">
    <xsd:element ref="cem:Equipment"/>
    <xsd:element ref="ccs:Error"/>
  </xsd:choice>
</xsd:complexType>
```

Figure 5 shows the information about AutoFOUP equipment returned from EDA Host, which is written in XML and interpreted by XMLSerializer.

```
<?xml version="1.0" encoding="UTF-8"?>
- <Equipment xmlns:xsi="http://www.w3.org/2001/XMLSchema-
  org:xsd.E120-1.V0310.CommonEquipmentModel">
     <Name>AFC01</Name>
     <Description>AFC01</Description>
     <Uid>33959483-0db4-47a0-8ecb-291b522bde07</Uid>
```

Figure 5. The description of AutoFOUP equipment written in XML

4. Conclusion

IoT is a distributed computing environment where the sensor and the object with control function are seamlessly connected with the user, and the interoperability support is essential for the data exchange between the services and applications. Service interface technology plays the role of connecting the IoT components with the Service and Applications.

In this paper, we design a software architecture based on SOA for EDA system of semiconductor manufacturing filed in IIoT environment. To design the software architecture which is used to integrate information systems in the factory, we analyze the requirements of EDA system in the prospect of the components of the information system including PLC, EDA Host, EDA client and EDA administrator in EDA SEMI standards which is aiming at improving interoperability and scalability and applying web services to the system. We showed the main characteristics of EDA based on web services that are useful to integrated heterogeneous service platform and diverse equipment in a factory of IIoT environment. As the result, standard techniques of web services like SOAP and WSDL give advantage to transform the traditional information systems with low level messaging to a well-designed systems to optimize operation of manufacturing in smart factory.

5. Acknowledgement

This work was supported by the research fund of Baekseok University (2016).

6. References

[1] Luigi Atzori, Antonio Iera, and Giacomo Morabito, “The Internet of Things: A survey,” Computer Networks: The International Journal of Computer and Telecommunications Networking, vol. 54, no. 15 pp. 2787-2805, 2010.
[2] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," Future Generation Computer Systems, vol. 29, no. 7, pp 1645-1660, 2013.
[3] Recommendations for implementing the strategic initiative INDUSTRIE 4.0, 2013.
[4] F. Tao, Y. Zuo, L. D. Xu, and L. Zhang, “IoT-Based intelligent perception and access of manufacturing resource toward cloud manufacturing,” IEEE Transactions on Industrial Informatics, vol. 10, no. 2, pp. 1547–1557, 2014.
[5] W3C, Web Services Architecture, http://www.w3.org/TR/2003/WD-ws-arch-20030808/
[6] W3C, Web Services Description Language(WSDL) 1.1, 2001, http://www.w3c.org/TR/wsdl
[7] W3C, Simple Object Access Protocol (SOAP) 1.1. 2000, http://www.w3c.org/TR/SOAP.
[8] F. Curbera et. al., "Unraveling the Web Services Web: An Introduction to SOAP, WSDL, and UDDI", IEEE Internet Computing, Vol. 6, No. 2, pp. 86-93, March/April 2002
[9] Sun, Java Message Service, http://java.sun.com/ products/jms/, 2001.
[10] S. Pallickara and G. Fox, "NaradaBrokering: a distributed middleware framework and architecture for enabling durable peer-to-peer grids", in Proceedings of the ACM/IFIP/USENIX 2003 International Conference on Middleware, Brazil, 2003.

[11] Pietzuch, P. R., Shand, and B., Bacon, J., "Composite Event Detection as a Generic Middleware Extension, " IEEE Network, vol. 18, pp. 44-55, 2004.
[12] François, A.R.J., Software Architecture for Computer Vision: Beyond Pipes and Filters, Prentice Hall, 2003.
[13] SEMI E120-1.V1104 specification for the common equipment model.
[14] SEMI E125-1.V0305 specification for equipment self-description.
[15] SEMI E132-1.V0305 specification for equipment client authentication and authorization.
[16] SEMI E134-1.V1105 specification for data collection management.