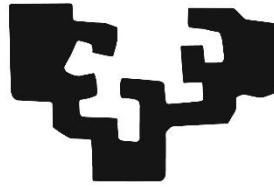


eman ta zabal zazu



Universidad del País Vasco Euskal Herriko Unibertsitatea
Escuela de Ingeniería de Bilbao

Departamento de Ingeniería de Sistemas y Automática

**METODOLOGÍAS PARA EL DISEÑO DE
SISTEMAS DE CONTROL DISTRIBUIDO
BAJO EL ESTÁNDAR IEC 61499
APLICADOS AL CONTROL DE PROCESOS**

Tesis Doctoral

Marcelo Vladimir García Sánchez
Tesis dirigida por Dr. Federico Pérez González

Enero 2018

Índice General

ÍNDICE GENERAL	II
ÍNDICE DE FIGURAS	V
ÍNDICE DE TABLAS	VII
NOMENCLATURA	VIII
CAPÍTULO I: INTRODUCCIÓN	1
1.1 MOTIVACIÓN	1
1.2 OBJETIVOS	6
1.3 HIPÓTESIS	7
1.4 METODOLOGÍA	9
1.5 ESTRUCTURA DE LA TESIS	10
1.6 APORTACIONES DE LA TESIS	10
CAPÍTULO II: ESTADO DEL ARTE	13
2.1 INTRODUCCIÓN	13
2.2 INDUSTRIA 4.0	15
2.3 INTERNET DE LAS COSAS (IoT)	19
2.3.1 <i>Internet Industrial de las Cosas (IIoT)</i>	20
2.4 COMUNICACIÓN MÁQUINA A MÁQUINA (M2M)	22
2.5 SISTEMAS CIBER-FÍSICOS (CPS)	23
2.5.1 <i>Sistemas de Producción Ciber-Físicos (CPPS)</i>	25
2.6 TECNOLOGÍAS DE COMUNICACIÓN EN LA AUTOMATIZACIÓN INDUSTRIAL.....	28
2.6.1 <i>Arquitectura Orientada a Servicios (SOA)</i>	28
2.6.2 <i>Servicios WEB</i>	30
2.6.2.1 SOAP	32
2.6.2.2 RESTful.....	33
2.6.3 <i>OPC-UA</i>	34
2.6.3.1 Servicios de Comunicación	39
2.6.3.2 Servicios Base	41
2.6.3.3 Modelo de Información.....	42

2.6.3.4 Modelo de Datos	45
2.6.3.5 Seguridad	47
2.7 ESTÁNDARES DE AUTOMATIZACIÓN DE SISTEMAS DISTRIBUIDOS	48
2.7.1 <i>Estándar IEC 61499</i>	49
2.7.1.1 Especificaciones IEC 61499	50
2.7.1.2 Arquitectura	50
2.7.2 <i>Entornos de Desarrollo y de Ejecución de la Norma IEC 61499</i>	56
2.7.2.1 FBDK / FBRT	57
2.7.2.2 4DIAC-IDE / FORTE	60
2.8 TRABAJOS RELACIONADOS	62
2.8.1 <i>Diseño de Sistemas CPPS bajo norma IEC 61499</i>	62
2.8.2 <i>Comunicaciones M2M para sistemas CPPS</i>	65
2.9 CONCLUSIONES	67

CAPÍTULO III: METODOLOGÍAS PARA DISEÑO DE CPPS 71

3.1 INTRODUCCIÓN	72
3.2 TRABAJOS RELACIONADOS	74
3.3 INTEGRACIÓN VERTICAL	77
3.4 ARQUITECTURA CPPS	80
3.4.1 <i>Modelo de Procesos de Producción</i>	81
3.4.1.1 Modelo de Topología de Planta	82
3.4.1.2 Modelo de Dispositivos Inteligentes de Planta	82
3.4.2 <i>Modelo de Intercambio de Información</i>	82
3.4.2.1 Servicios Atómicos	83
3.4.2.2 Nodos Lógicos de Proceso	83
3.4.2.3 Dispositivos Lógicos CPS	83
3.4.3 <i>Modelo de Información de Planta</i>	83
3.5 IEC 61499 COMO HERRAMIENTA PARA INTEGRACIÓN DE ARQUITECTURAS CPPS	84
3.5.1 <i>Elementos básicos de un SIFB</i>	85
3.5.2 <i>Especificaciones del SIFB</i>	87
3.5.3 <i>Metodología de Diseño de SIFB</i>	91
3.5.4 <i>Implementación de la metodología en 4DIAC</i>	93
3.6 MODELADO DE LA ARQUITECTURA CPPS BAJO LA NORMA IEC 61499	95
3.7 OPC-UA E IEC 61499 PARA INTEGRACIÓN DE COMUNICACIONES EN ARQUITECTURAS CPPS	96
3.7.1 <i>Conjunto de SIFBs para OPC-UA</i>	96
3.7.1.1 SIFB OPCUA_SERVER	96
3.7.1.2 SIFB SERVER OPC-UA Dinámico	100
3.7.1.3 SIFB OPCUA_CLIENT_READ	102
3.7.1.4 SIFB OPCUA_CLIENT_WRITE	103
3.7.1.5 SIFB OPCUA_CLIENT_SUBSCRIBE	104
3.8 CASO DE ESTUDIO	105
3.9 CONCLUSIONES	108

CAPÍTULO IV: MODELADO DE APLICACIONES DISTRIBUIDAS BAJO ISA 88/95	111
4.1 TRABAJOS RELACIONADOS	112
4.2 ESTÁNDARES INDUSTRIALES.....	115
4.2.1 Norma ISA 95	115
4.2.2 Norma ISA 88	117
4.3 PROPUESTA DE META-MODELO DE TOPOLOGÍA DE PLANTA (PTM) BASADOS EN ISA 88/95.....	118
4.4 META-MODELO DE DISPOSITIVOS INTELIGENTES DE PLANTA (PIDM)	122
4.5 CASO DE ESTUDIO MODELADO ISA 88/95	123
4.6 CREACIÓN DE LA HERRAMIENTA PARA GENERACIÓN DE APLICACIONES DISTRIBUIDAS	129
4.6.1 Trabajos Relacionados	130
4.7 DESARROLLO DE HERRAMIENTA DE INGENIERÍA.....	132
4.7.1 Perfil de Conformidad para verificación de la factibilidad de software basados en IEC 61499	132
4.7.2 Diseño de software para la creación de archivos de configuración basados en IEC 61499 ..	135
4.8 CASO ESTUDIO HERRAMIENTA DE INGENIERÍA SOFTWARE	142
4.9 CONCLUSIONES.....	143
CAPÍTULO V: CONCLUSIONES Y FUTURAS LÍNEAS DE INVESTIGACIÓN	147
5.1 CONCLUSIONES Y PRINCIPALES APORTACIONES	147
5.2 FUTURAS LÍNEAS DE INVESTIGACIÓN	152
BIBLIOGRAFÍA.....	155

Índice de Figuras

Figura 1: Historia de Evolución Industrial.....	16
Figura 2: Visión del Internet Industrial de las Cosas (IIoT)	22
Figura 3: Arquitectura de Automatización Industrial.....	27
Figura 4: Automatización Industrial en ambiente SOA	29
Figura 5: Tecnología de OPC-UA.....	35
Figura 6: Arquitectura SOA en Automatización	36
Figura 7: Pilares OPC-UA.....	37
Figura 8: Partes de OPC-UA	38
Figura 9: Perfiles de Transporte de OPC-UA.....	39
Figura 10: Conjuntos de Servicios de OPC-UA.....	41
Figura 11: Comparación entre modelos OPC y OPC-UA.....	43
Figura 12: Capa de Modelos de OPC-UA	44
Figura 13: Resumen de los principales nodos en OPC-UA	45
Figura 14: Concepto de Seguridad Escalable.....	48
Figura 15: Modelo de Bloque de Función	51
Figura 16: Modelo de recurso	53
Figura 17: Modelo de dispositivo	54
Figura 18: Modelo de sistema	54
Figura 19: Modelo de aplicación	55
Figura 20: Modelo de distribución	56
Figura 21: Sistema de configuración en FBDK.....	59
Figura 22: Sistema de configuración en 4DIAC-IDE.....	61
Figura 23: Arquitectura General para dispositivos CPPS.....	81
Figura 24: Estructura conceptual del IEM	84
Figura 25: Estructura General de un SIFB.....	86
Figura 26: Tipos de Servicio de SIFB Requester y Responder.....	88
Figura 27: Bloques de interfaz de comunicaciones Publish y Subscribe.....	89
Figura 28: Bloques de interfaz de comunicaciones Client y Server.....	90
Figura 29: Elementos en el desarrollo de SIFBs en C++.....	91

Figura 30: Escenario general de desarrollo del software en 4DIAC	93
Figura 31: Estructura en C++ de SIFB OPCUA_SERVER.....	94
Figura 32: Algoritmo en C++ para integración en FORTE de SIFB OPCUA_SERVER	94
Figura 33: Modelo de implementación de servidor OPC UA	97
Figura 34: Meta-Modelo del Espacio de Direcciones.....	98
Figura 35: Modelo de Dispositivos de Campo	99
Figura 36: Ejemplo de Archivo de Configuración de servidor OPC UA en formato XML	99
Figura 37: SIFB OPCUA_SERVER.	100
Figura 38: Arquitectura implementación SIFB de servidor OPC UA.....	101
Figura 39: SIFB SERVER OPC UA dinámico.....	102
Figura 40: SIFB OPCUA_CLIENT_READ	103
Figura 41: SIFB OPCUA_CLIENT_WRITE.....	103
Figura 42: SIFB OPCUA_CLIENT_SUBSCRIBE.....	104
Figura 43: Plano P&ID del Sistema ESP y cabeza de pozo (xmas wellhead)	105
Figura 44: Sistema de comunicación simulado para el proceso de separación y extracción de crudo	106
Figura 45: Componente de Información (ICs) de Pozo PAA001.....	107
Figura 46: Modelo intercambio de información	116
Figura 47: Sintaxis del Meta-Modelo de Topología de Planta	119
Figura 48: Caracterización de los Transmisores	121
Figura 49: Meta-modelo mínimo para los dispositivos inteligentes capaces de suministrar datos.....	122
Figura 50: Proceso de Producción de Crudo	123
Figura 51: Meta - modelo de Topología de Planta de Petroamazona B18	125
Figura 52: Instrumentación separador de crudo V1401	127
Figura 53: Ejemplo de modelo de Topología de Planta.....	128
Figura 54: Ejemplo de modelo de Dispositivos	129
Figura 55: Metamodelo servicio REQUEST.....	133
Figura 56: Metamodelo servicio RESPONSE.....	134
Figura 57: Modelo General Herramienta Software.....	136
Figura 58: Metamodelo de la Herramienta Software	138
Figura 59: SIFB LOG DATA.....	139
Figura 60: Función FUNCTION BLOCK de Herramienta Software	140
Figura 61: Función SEND de Herramienta Software	141
Figura 62: Arquitectura de comunicación Separador de Crudo V1401	142
Figura 63: Archivo de configuración XML para crear servidores OPC UA.....	143

Índice de Tablas

Tabla 1: Comparación de los principios para diseñar componentes en Industria 4.0 ...	19
Tabla 2: Arquitectura 5C para la implementación de CPPS	25
Tabla 3: Sistemas de Desarrollo y Ejecución de IEC 61499	57
Tabla 4: Eventos y datos de entrada estándar de un SIFB	86
Tabla 5: Eventos y datos de salida estándar de un SIFB.....	87
Tabla 6: Entradas y salidas de un FB de comunicaciones.....	88
Tabla 7: Relación entre los modelos de la arquitectura CPPS e IEC 61499.....	95

Nomenclatura

4DIAC	Framework for Distributed Industrial Automation & Control, Marco para la Automatización y Control Industrial Distribuido
AML	Automation Markup Language, Lenguaje de Marcas para Automatización
API	Application Programming Interface, Interfaz de Programación de Aplicaciones
AS	Atomic Services, Servicios Atómicos
CAEX	Computer Aided Engineering Exchange, Ingeniería de Intercambio asistida por computador
CM	Control Module, Módulo de Control
COM/DCOM	Component Object Model/Distributed COM, Modelo de Objetos de Componentes/COM Distribuidos
CPFD	Compliance Profile for Feasibility Demonstrations, Perfil de cumplimiento para la demostración de factibilidad
CPPS	Cyber Physical Production Systems, Sistemas de Producción Ciber-Físicos
CPS	Cyber Physical Systems, Sistemas Ciber-Físicos
CPSLD	CPS Logical Devices, Dispositivos Lógicos CPS
DCS	Distributed Control System, Sistema de Control Distribuido
DPWS	Devices Profile for Web Services, perfiles de Dispositivos para Servicios Web
DTM	Dynamic synchronous transfer mode, Modo de transferencia dinámica síncrona
EDDL	Electronic Device Description Language, Lenguaje para descripción electrónica del equipo
EM	Equipment Module, Modulo de Equipo
ERP	Enterprise Resource Planning, Planificación de Recursos Empresariales

FB	Function Block, Bloque de Función
FBD	Function Block Diagram, Diagrama de bloque de funciones
FDI	Field Device Integration, Integración de Dispositivos de Campo
FORTE	4DIAC Runtime Environment, Ambiente de ejecución para 4DIAC
HMI	Human Machine Interface, Interfaz Hombre - Máquina
HTTP	Hypertext Transfer Protocol, Protocolo de Transferencia de Hipertexto
HTTPS	Hypertext Transfer Protocol Secure, Protocolo seguro de Transferencia de Hipertexto
IC	Information Components, Componentes de Información
IDE	Integrated Development Environment, Entorno de Desarrollo Integrado
IEM	Information Exchange Model, Modelo de Intercambio de Información
IIoT	Industrial Internet of Things, Internet Industrial de las Cosas
IL	Instruction List, Lista de Instrucciones
IoD	Internet of Devices, Internet de los Dispositivos
IoS	Internet of Services, Internet de los Servicios
IoT	Internet of Things, Internet de las Cosas
IP	Internet protocol, Protocolo de Internet
ISA	International Society of Automation, Sociedad Internacional de Automática
LD	Ladder Diagram, Lenguaje escalera
LPN	Logical Process Node, Nodo de Proceso Lógico
M2M	Machine to machine communication, Comunicación Máquina a Máquina
MES	Manufacturing Execution System, Sistema de Ejecución de Manufactura
MVCC	Multiversion concurrency control, Control de Concurrencia mediante Versiones Múltiples
ODBC	Open DataBase Connectivity, Conectividad abierta de bases de datos
OLE	Object Linking and Embedding, Incrustación y Enlazado de Objetos
OPC	OLE for Process Control, OLE para Procesos de Control

OPC UA	Open Platform Communications Unified Architecture, Plataforma de comunicación abierta con Arquitectura Unificada
PIDM	Plant Intelligent Devices Model, Modelo de Dispositivos Inteligentes de Planta
PIM	Plant Information Model, Modelo de Información de Planta
PLC	Programmable Logic Controller, Controlador lógico programable
PPM	Production Process Model, Modelo de Procesos de Producción
PTM	Plant Topology Model, Modelo de Topología de Planta
PU	Production Unit, Unidad de Producción
SCADA	Supervisory Control And Data Acquisition, Supervisión, Control y Adquisición de Datos
SFC	Sequential Function Chart, Bloques de función secuenciales
SIFB	Service Interface Function Block, Bloque de Función de Interfaz de Servicio
SOA	Service Oriented Architecture, Arquitectura orientada a servicios
SOAP	Simple Object Access Protocol, Protocolo de Acceso simple a Objetos
SQL	Structured Query Language, Lenguaje de Consulta Estructurada
SSN	Semantic Sensor Network, Red de Sensores Semánticos
ST	Structured Text, Texto estructurado
TCP	Transfer Control Protocol, Protocolo de Control de Transferencia
UML	Unified Model Language, Lenguaje de Modelado Unificado
WS*	Web Services, Servicios Web
WU	Work Unit, Unidad de Trabajo
XML	eXtensible Markup Language, Lenguaje de Marcas Extensible

1

INTRODUCCIÓN

1.1 MOTIVACIÓN

La producción de petróleo del Ecuador actualmente está disminuyendo, los precios mantienen su tendencia a la baja y los costes de extracción y operación no han cambiado conforme ha decrecido la producción, en consecuencia, es pertinente llevar a cabo iniciativas para reducción de gastos en el sector *upstream* de la industria petrolera ecuatoriana. Es por esta razón que las empresas de producción y exploración de crudo en el Ecuador y a nivel mundial están inmersas en lo que la 4^{ta} revolución industrial ha llamado *transformación digital* como un paso para aumentar la eficiencia en los procesos. Teniendo en cuenta esta definición, se puede inferir que la transformación digital de este sector industrial está estrechamente relacionada a lo que se denomina *Industria 4.0* [1] la que permitirá que la industria petrolera posea arquitecturas flexibles para procesar diferentes tipos de crudo, automatización a bajo coste y optimización de los recursos generando beneficios tanto a nivel de proceso, como de producto y de modelo de negocio.

Al hablar de Industria 4.0, uno de los conceptos que más relevancia ha adquirido es el de Internet de las Cosas (IoT por sus siglas en inglés, acrónimo de Internet of Things).

El concepto de IoT es importante para la transformación digital de la industria. Como concepto general no es nuevo, internet fue desde el principio una red de redes que permitía conectar computadoras para compartir datos. El término IoT se utilizó por primera vez en 1999, refiriéndose a objetos identificables (things) y su representación virtual en una infraestructura digital. En su definición sintética, IoT es una red que conecta los mundos físicos (*dispositivos*) y virtuales (*sistemas*), en los que millones de dispositivos y sistemas colaboran entre ellos y con otros para proporcionar servicios inteligentes (*smart*) a los usuarios. Si bien el concepto no es nuevo, está cobrando cada vez mayor relevancia en la industria y en este campo se le ha conceptualizado como Internet Industrial de las Cosas (IIoT), se trata de IoT donde las cosas a interconectar son las máquinas, personas y objetos dentro de del entorno industrial.

Dentro de la Industria 4.0 otra tecnología habilitadora que juega un papel importante es la comunicación máquina a máquina (M2M por sus siglas en inglés, acrónimo de Machine to Machine). M2M se considera a cualquier tecnología que permita, a los dispositivos que se encuentren en una misma red, intercambiar información y realizar acciones de forma totalmente autónoma. Esta forma de comunicación se utiliza principalmente para la supervisión remota de la propia máquina y del entorno en el que se encuentra; esto sería de gran ayuda en sector upstream de la industria petrolera debido a que los lugares de extracción se encuentran distribuidos a lo largo de toda la región amazónica del Ecuador.

Debemos considerar a M2M como el nivel base de la comunicación, mientras que IIoT está utilizando esa comunicación para proporcionar una solución, como puede ser en el ámbito industrial o incluso en el personal. A corto plazo, incluso aquellos que buscan agresivamente la Industria 4.0, tendrán una mezcla de dispositivos inteligentes y equipos con productos tradicionales y máquinas que necesitan la comunicación M2M para guiar tanto el proceso de integración vertical como el horizontal. [2]

Una tercera tecnología habilitadora en la cual la Industria 4.0 se basa son los Sistemas Ciber Físicos (CPS por sus siglas en inglés, acrónimo de Cyber Physical System). CPS es el resultado de dotar a los componentes u objetos físicos que nos encontramos de

forma habitual en nuestro entorno de trabajo, de capacidades de computación y de comunicación para convertirlos en objetos inteligentes. Estos permiten así superar a los simples sistemas empotrados actuales en cuanto a capacidad, seguridad, escalabilidad, adaptabilidad, resiliencia y usabilidad, pudiendo trabajar en conjunto formando ecosistemas distribuidos y totalmente autónomos.

Los CPS van más allá del concepto de IoT ya que se refiere a sistemas de mayor complejidad compuestos a partir de otros sistemas y que son capaces de aprender de las interacciones que tienen con el mundo físico, de forma que convierten los entornos en inteligentes. Este concepto aplicado a la industria se ha llamado Sistemas Ciber Físicos de Producción (CPPS acrónimo de Cyber Physical Production System). Los CPPS están constituidos por dispositivos industriales de control con amplias capacidades de computación y comunicación, tanto local como remota.

Por otra parte, la norma de automatización IEC 61499 promueve un marco para el desarrollo basado en modelos para sistemas de control distribuidos. Dispositivos CPPS en unión con IEC 61499 permitirá modelar y desarrollar componentes software y hardware para sistemas de control distribuidos facilitando la recopilación y el procesamiento de datos a nivel de sensores y actuadores, optimizando las técnicas de mantenimiento predictivo de equipos y permitiendo la incorporación de nuevas técnicas de control en el sector upstream de la industria del petróleo y gas. [3]

Para desarrollar los componentes software distribuidos IEC 61499 provee el modelo de Bloque de Función de Interfaz de Servicio (SIFB), que permite encapsular y abstraer al usuario del acceso al hardware, a las comunicaciones o a los recursos de las Interfaces de Programación de Aplicaciones (API).

La norma IEC 61449 brinda las siguientes características: [4]

- **Portabilidad:** soportar e interpretar correctamente configuraciones y componentes software creadas por diferentes herramientas software.
- **Interoperabilidad:** los distintos dispositivos integrados pueden funcionar conjuntamente para llevar a cabo las funciones propias de las aplicaciones

distribuidas.

- **Configurabilidad:** cualquier dispositivo y sus componentes software pueden ser configurados por herramientas de software de múltiples proveedores.
- **Reconfigurabilidad:** supone la habilidad para adaptar el hardware y el software de control durante la operación del proceso.
- **Distribución:** permite distribuir componentes software en diferentes dispositivos hardware independientemente del proveedor, el cual, es un requisito necesario dado por la industria de la automatización.

El uso de estándares como IEC 61499 y arquitecturas CPPS implicaría un ahorro considerable en el hardware de automatización usado por la industria de petróleo ecuatoriana debido a que se podrá implementar este tipo de tecnologías en dispositivos de bajo coste [5]. La automatización de bajo coste posibilita que nuevas tecnologías puedan ser incorporadas rápidamente y con un bajo impacto en los presupuestos de las empresas, la principal ventaja es el uso de hardware de diversos fabricantes, optimización de la producción e incorporación de una mayor flexibilidad e individualización de la automatización de los procesos de extracción y producción de petróleo.

Otro problema que tiene la industria de producción de petróleo, tanto en el caso upstream como en el midstream y downstream, es que se caracteriza por estar formada por una estructura distribuida en múltiples ubicaciones geográficas muy distantes entre sí, como se mencionó en párrafos anteriores. Por tanto, existe una clara necesidad de mejorar la gestión del proceso distribuido, adaptando el sistema de control a las características propias de múltiples ubicaciones para las partes del proceso, y ofreciendo la posibilidad de monitorizar remotamente y dar respuesta a las demandas de producción en tiempo real.

Para alcanzar esto es necesario contar con la posibilidad de monitorizar remotamente el estado y evolución del proceso de deshidratación de crudo, transporte y refinación del mismo. En este sentido, la industria de producción de petróleo podría beneficiarse adicionalmente de otras propuestas impulsadas desde Industria 4.0 [6]. Esta iniciativa,

como se explicó en párrafos anteriores, promueve, la introducción de conceptos en fábrica como son el IIoT el cual define un conjunto de tecnologías para el acceso remoto al sistema productivo y el Internet de Servicios (IoS), que proporcionan mecanismos para la fabricación basada en la nube y virtualización de procesos. Éstas iniciativas, tendrán un impacto colectivo en todos los aspectos de las empresas de fabricación y se hace conveniente introducir estos conceptos en industrias reales como es la del sector de petróleo y gas. [7]

Industria 4.0 insiste en la importancia de utilizar estándares consolidados para representar formalmente la información que se captura, guarda y transmite, así como para especificar las tecnologías de comunicación para su transmisión. En este contexto, la norma OPC-UA (IEC 62541) es una alternativa prometedora para implementación de comunicaciones M2M ya que cuenta con una arquitectura orientada a servicios que al mismo tiempo ofrece seguridad de datos y modelos de información fiables. Por desgracia, y quizás debido a la falta de disponibilidad de plataformas adecuadas, esta tecnología aún no se encuentra ampliamente utilizada a nivel industrial. [8]

OPC-UA utiliza archivos de configuración para que los servidores realicen la integración de los datos a nivel de planta. Estos archivos de configuración deben obligatoriamente seguir modelos de planta y de los dispositivos que suministran los datos. Dentro de las normas usadas ampliamente a nivel industrial tenemos el estándar ISA95 (IEC 62264) y el modelo físico de ISA88 (IEC 61512) los cuales proponen un modelo jerárquico donde se caracteriza el rol que toma el equipamiento en los procesos de producción. El uso de estas normas facilitará que los archivos de configuración inicial de servidores OPC-UA sigan un modelo de información aceptado a nivel industrial permitiendo una integración rápida de esta tecnología de comunicación en el sector de la industria petrolera ecuatoriana.

Considerando las necesidades identificadas en el sector de producción del petróleo, la motivación de este trabajo es tender puentes entre el estado de la tecnología y el diseño de aplicaciones de control industrial a través del establecimiento de metodologías y la integración de una plataforma CPPS para su uso en control distribuido bajo el estándar

IEC 61499 en dispositivos empotrados de bajo costo, y una arquitectura OPC-UA fácilmente configurable, con servidores y clientes gestionados desde aplicaciones IEC 61499 y con archivos de configuración que sigan meta-modelos basados en ISA 88/95. Por lo tanto, la propuesta presentada permite una plataforma de software fácilmente desplegable destinada a recoger remotamente datos de proceso y que posibilita la integración vertical desde las áreas de empresa o negocio hasta el nivel de producción en planta.

1.2 OBJETIVOS

La investigación se centrará en la idea de unir los conceptos de componentes funcionales de información y de aplicaciones de control distribuidas, utilizando siempre estándares de aplicación industrial.

El objetivo principal de la presente tesis es: *Definir una plataforma que permita la integración de una arquitectura industrial basada en CPPS para control distribuido bajo norma IEC 61499 mediante la utilización de dispositivos empotrados de bajo coste.*

En base a estas observaciones y con el fin de lograr el objetivo principal, se plantean cuatro objetivos parciales:

- Definir metodologías para el desarrollo de módulos de control y de comunicación conforme al estándar IEC 61499 que formarán parte de un repositorio de SIFBs.
- Integrar la plataforma IEC 61499 en un sistema empotrado de bajas prestaciones y un entorno de desarrollo de libre distribución. Para lograr esto, se desarrollará nuevos FBs que junto con los ya existentes en el estándar permitan la creación de aplicaciones de control que posteriormente se mapearán en los diferentes recursos hardware para formar una plataforma CPPS. A pesar de que durante los últimos años una gran cantidad de investigadores han estado trabajando en el desarrollo y evolución del estándar IEC 61499, aún queda un largo camino por recorrer para su adopción por la industria ya que son necesarias aún más modificaciones y ampliaciones a la norma con el fin de ser utilizado con eficacia en el contexto de los sistemas de control distribuidos de automatización industrial.

- Implementar servidores y clientes OPC-UA configurable bajo norma IEC 61499 para comunicaciones con protocolos industriales tradicionales a nivel de planta. El enfoque se centra en la integración de protocolos comúnmente usados a nivel industrial, para tener acceso a datos de proceso y realizar una integración vertical eficiente aplicado a la industria del petróleo y gas. La configuración inicial de servidores OPC-UA seguirán meta-modelos basados en ISA 88/95 para integrarlos dentro de arquitecturas flexibles para CPPS usadas en la industria.
- Diseñar e implementar un Entorno de Desarrollo Integrado (IDE- Integrated Development Environment) que dé soporte al diseño y desarrollo de aplicaciones M2M mediante el uso FBs de servidores y clientes OPC-UA para implementar una plataforma CPPS de control industrial distribuidas, de complejidad creciente, con un alto grado de reutilización y fáciles de mantener y extender. El IDE dará soporte al ciclo de desarrollo de la aplicación a través de la integración de herramientas de dominio. Por lo tanto, debe embeber metodologías que guíen al diseñador, así como herramientas que faciliten el diseño de la aplicación. Para el diseño del IDE se evaluará la aplicabilidad de tecnologías y metodologías consolidadas en otros campos de aplicación y, en la medida de lo posible, se integrarán de forma transparente al usuario, de forma que se evite, hasta cierto punto, el continuo reciclaje del usuario final en tecnologías, metodologías y herramientas de complejidad creciente.

1.3 HIPÓTESIS

Los problemas del sector upstream de la industria petrolera ecuatoriana se pueden resumir en: automatización con hardware costoso, bajo uso de normas de modelado de información ampliamente usados en otros campos industriales y una deficiente integración vertical de los procesos desde el nivel de planta hasta el nivel de administración. En busca de una solución a estos problemas se puede plantear la siguiente hipótesis principal:

El desarrollo de CPPSs para integración de comunicaciones a nivel de planta basados en el protocolo OPC-UA permite incrementar la escalabilidad de la norma IEC 61499 y su uso en la industria del petróleo y gas.

Esta hipótesis se basa en trabajos desarrollados por [9]–[11] en los cuales se usa tecnologías propuestas dentro de la Industria 4.0 como son los CPPS en unión con el protocolo OPC-UA demostrando que es posible mejorar la recopilación de datos del proceso de producción y exploración de crudo, pero no se han enfocado en uso de normas para sistemas distribuidos como es IEC 61499 desarrollada específicamente para este tipo de automatización.

Mientras que las empresas de producción y exploración de crudo gastan millones de dólares en infraestructuras complejas de automatización; la captura, la centralización y la difusión de los datos de estos sistemas en toda la empresa es muy difícil. Parte del desafío proviene del uso de múltiples protocolos propietarios, ancho de banda de la red bajo, conexiones poco fiables en lugares remotos, transferir los datos de proceso en las aplicaciones del negocio, y la navegación por las diversas arquitecturas de TI.

Afortunadamente, la tecnología de comunicación implementada por OPC-UA hace posible que la captura de datos sea fiable, permite almacenar y reenviar los datos en un formato común para poder transferirlos entre las aplicaciones de automatización, control y bases de datos enfocadas a los negocios de una manera sencilla y establece conexiones seguras a través de redes informáticas complejas. Por lo tanto, se tendrá una segunda hipótesis que permitirá solucionar este inconveniente detectado:

El uso de OPC-UA como tecnología habilitadora para el intercambio de datos a nivel de planta, implementando de esta manera conceptos dados por IIoT y M2M, permite mejorar la disponibilidad y confiabilidad de la información en el proceso upstream de la industria petrolera ecuatoriana.

Esta segunda hipótesis para su comprobación necesita definir un esquema de comunicación M2M mediante el uso de servidores y clientes OPC-UA configurables de manera estática y dinámica en unión con los modelos dados por la norma IEC 61499.

Por último, las estrategias de comunicación basadas en IEC 61499 y OPC-UA diseñadas e implementadas tendrán que ser validadas experimentalmente, lo que requerirá el diseño de una plataforma de automatización de bajo coste adecuada a las necesidades que plantean la industria de explotación y producción de petróleo haciendo énfasis en la industria petrolera del Ecuador específicamente en la empresa Petroamazonas EP de Ecuador.

1.4 METODOLOGÍA

Los métodos de investigación científica que intervienen para la consecución de la presente tesis doctoral son el método investigativo y el experimental. El primer método permite la recopilación de la información necesaria para poder iniciar el desarrollo de la investigación, y una vez cimentados los conocimientos acerca del tema, proceder con el desarrollo de SIFBs bajo IEC 61499 con integración de protocolo OPC-UA. Con la ayuda del segundo método se verifica que lo diseñado y programado funcione correctamente en casos de estudios que simulen procesos industriales de producción de crudo y gas.

El orden cronológico del procedimiento es como sigue:

- Revisión exhaustiva del estándar IEC 61499 para aplicaciones de control distribuidas, en el cual se fundamentará la metodología de diseño de componentes de información.
- Investigación de las diferentes alternativas en el diseño de componentes de información.
- Análisis de las herramientas software disponibles para el diseño de aplicaciones de control distribuidas.
- Diseño y desarrollo de una metodología de diseño de sistema distribuido conforme al estándar IEC 61499 independiente de los sistemas de comunicación.
- Diseño e implementación de componentes de información: A partir de la metodología planteada y utilizando las herramientas disponibles, realizar un conjunto de bloques funcionales SIFBs para OPC-UA con capacidad de integración con diferentes redes de comunicación industrial.

- Implementación de una herramienta capaz de generar aplicaciones de control distribuidas basada en componentes de información conforme al estándar IEC 61499 mediante la introducción automática de SIFBs.

1.5 ESTRUCTURA DE LA TESIS

Una vez definidos la motivación y los objetivos del trabajo de investigación, el capítulo 2 revisa brevemente el Estado del Arte de la teoría de la evolución de los sistemas de automatización y analiza las tecnologías de comunicación aplicables para el paradigma de la industria 4.0. Además, se comentan los trabajos de otros autores relacionados con los objetivos comentados.

A lo largo del capítulo 3 se presenta una arquitectura abierta para la integración vertical basada en sistemas CPPS, configurados bajo la norma IEC 61499 y usando OPC-UA, apta para su utilización en la fabricación flexible en la industria de petróleo y gas.

En el capítulo 4 se desarrollan meta-modelos para generar archivos de configuración para servidores OPC-UA a partir de los modelos de la planta y de los dispositivos que suministran los datos basados en la norma ISA95/88, posteriormente se indica una herramienta de desarrollo integrado (Integrated Development Environment – IDE) la cual permite crear aplicaciones IEC 61499 de una manera simple y sencilla que permitirá la integración rápida de esta norma en ambientes industriales. Se ha realizado la validación de estos modelos y herramienta en parte del proceso que desarrolla la compañía Petroamazonas EP de Ecuador.

Finalmente, en el capítulo 5 se agrupan las conclusiones de la tesis y los posibles trabajos futuros relacionados.

1.6 APORTACIONES DE LA TESIS

Este proyecto de tesis está orientado a generar un conocimiento práctico que sea válido para extender los conceptos de Industria 4.0 e IEC 61499 dentro del campo de la industria de exploración y producción de crudo no sólo en el Ecuador si no a nivel mun-

dial, específicamente dando una respuesta al reto de portabilidad y configurabilidad dentro de los CPPS distribuidos partiendo de una aproximación basado en IEC 61499 que pueda ser extensibles a otras industrias.

Tal como lo detallan los objetivos de esta tesis doctoral se analiza, diseña, desarrolla y valida una metodología genérica para implementación de CPPS distribuidos que permite obtener una aplicación basada en SIFBs que integra comunicaciones a nivel industrial mediante el uso del protocolo OPC-UA como mecanismo de interoperabilidad y soporte en el proceso de diseño y desarrollo de aplicaciones distribuidas a nivel industrial.

Otro aporte que pretende dar esta Tesis Doctoral se fundamenta en un tema no abordado hasta el momento por otros investigadores y es dar soporte a la generación automática de SIFBs para la creación de Componentes de Información usando un Entorno de Desarrollo Integrado (IDE) genérico válido para cualquier aplicación distribuida sensible al contexto de cualquier dominio; especialmente centrado en la industria del procesamiento de petróleo y gas. Sin embargo, al mismo tiempo también puede ser personalizada a las particularidades concretas de un determinado campo de aplicación, e incluso incorporar nuevos requisitos de otros campos industriales no contemplados.

Como consecuencia este IDE permitirá introducir flexibilidad en las aplicaciones distribuidas basadas en IEC 61499, aportándoles fundamentalmente: i) *ejecución distribuida*, siendo posible seleccionar los nodos que alojen las instancias de comunicación en cada momento dentro de una arquitectura CPPS. ii) *mayor disponibilidad*, que además es transparente a la aplicación. No sólo debido a una gestión flexible de los recursos IEC 61499 disponibles, sino también debido a la posibilidad de asegurar la consistencia de su estado tras un proceso de descarga de la aplicación. iii) *escalabilidad*, tanto de aplicaciones como de recursos.

En resumen, las aportaciones son:

1. Metodología de diseño de CPPSs para su uso en entornos industriales de alta complejidad como es la explotación de crudo y gas.
2. Entorno de Generación de Aplicaciones distribuidas genérico (IDE) para la

creación de componentes de información para integración de comunicaciones industriales basados en el uso de OPC-UA bajo IEC 61499.

2

ESTADO DEL ARTE

2.1 INTRODUCCIÓN

La Industria 4.0 se fundamenta en la aplicación masiva de nuevas tecnologías de comunicación, con Internet a la cabeza, a todos los procesos de fabricación, de modo que el funcionamiento de la misma sea inteligente y eficiente. Con estas premisas podríamos abordar una definición más técnica, en virtud de la cual la Industria 4.0 consistiría en la implementación de una red tecnológica de producción inteligente para que máquinas, dispositivos y sistemas colaboren entre sí.

Los conceptos dados permitirán una producción dinámica y la incorporación de una mayor flexibilidad e individualización de los procesos de fabricación. Es por esta razón que todos los sistemas deben estar en capacidad de comunicarse directamente entre sí: los sensores, chips de comunicación, controladores, PLCs, HMIs, Sistemas de Ejecución de Producción (MES) y Sistemas de Planificación de Recursos Empresariales (ERP), proporcionando datos de producción importantes y en tiempo real para la toma de decisiones en las empresas. [12]

Las solicitudes de datos en arquitecturas de control convencionales son de dos tipos: por eventos o iniciados cíclicamente. Los datos siempre se obtienen en respuesta a las peticiones de sistemas de alto nivel, es decir, desde el nivel del cliente. El nivel inferior siempre actúa como un servidor y responde. Con la Industria 4.0, esta estricta separación de niveles y el enfoque de integración vertical comenzarán a desaparecer. En una red inteligente, cada dispositivo o servicio pueden iniciar de forma autónoma la comunicación con otros servicios.

En la actualidad las empresas necesitan solventar procesos complejos de producción, comunicación e integración. Son aplicaciones como ERP, MES, SCADA y otras que necesitan de integración vertical de manera rápida y eficiente. Generalmente, los enfoques comunes para la integración vertical generalmente separan los Sistemas de Automatización en diferentes capas (paradigma CIM) [13]. La estructura resultante se denomina pirámide de automatización y aunque se trata de un enfoque viable y usado a nivel industrial desde hace años, esto generalmente significa que todas las solicitudes de comunicación (por ejemplo, invocación de funciones, adquisición de datos, etc) tienen que ser transferidos de una capa a otra; y por lo tanto son susceptibles a errores cada vez que pasa a través de una interfaz propietaria con todas las traducciones de datos asociados que deben realizarse. Es por esto que la comunicación OPC Arquitectura Unificada (OPC-UA) proporciona una adecuada vía para integrar sistemas de gestión de procesos industriales directamente con el nivel de control, todo a través de una interface común.

OPC-UA es una nueva norma para comunicaciones en aplicaciones industriales desarrolladas por la Fundación OPC. OPC-UA contiene varias mejoras importantes con respecto a OPC clásica, por ejemplo, arquitectura orientada a servicios, seguridad de datos y modelos de información configurables.

La configurabilidad de los modelos de información hace que sea posible utilizar modelos de datos estandarizados, por ejemplo, IEC 61499, ISA 95 e ISA 88, junto con OPC-UA. Sin embargo, todavía no hay muchas experiencias de cómo las normas mencionadas deben utilizarse con OPC-UA y cuáles serían los beneficios reales de éste.

El presente capítulo está organizado de la siguiente manera: una breve descripción del Estado de Arte referente a Industria 4.0, Internet Industrial de las Cosas, Sistemas de Producción Ciber-Físicos. A continuación, se presenta algunas tecnologías de comunicación usadas en la automatización industrial como son Sistemas SOA, Servicios Web y OPC-UA, estandarizado como IEC 62541. Finalmente se presenta la norma IEC 61499 desarrollada para control de Sistemas Distribuidos, y trabajos relacionados con los objetivos de esta tesis.

2.2 INDUSTRIA 4.0

La digitalización es un importante mercado en crecimiento y muy atractivo a nivel industrial. El objetivo es fomentar la integración de las TI con productos, sistemas, soluciones y servicios a través de la cadena de valor que va desde el diseño de productos hasta el mantenimiento de los equipos. Adicionalmente siguen apareciendo nuevas oportunidades de negocios a nivel industrial como la digitalización de productos y sistemas, nuevas y mejoradas soluciones de software y nuevos servicios digitales. [14]

A fin de mantener la competitividad en los países industrializados es preciso afrontar nuevos retos que permitan incrementar la eficiencia en la fabricación de productos, con ciclos de vida cada vez más cortos, mediante un uso eficaz de energía y recursos. También es necesario reducir los tiempos de comercialización, producir productos más complejos con ciclos de innovación cada vez más rápidos e incrementar la flexibilidad de la producción utilizando sistemas de producción en masa e individualizada.

Industria 4.0 tiene por objetivo la implementación de la llamada fábrica inteligente, que no solo hace uso de datos provenientes de la interconexión de máquinas y de sistemas al interior del proceso de producción, sino con información externa al proceso provista por la oferta y la demanda del mercado, y por otras fábricas inteligentes, con el propósito de lograr una mayor adaptabilidad a las necesidades del mercado y a la producción industrial, lo cual conlleva a una asignación más eficaz de los recursos.

El fin de Industria 4.0 es impulsar activamente la reconfiguración de la industria para ayudar a crear un nuevo ambiente de trabajo de productividad integrada entre el

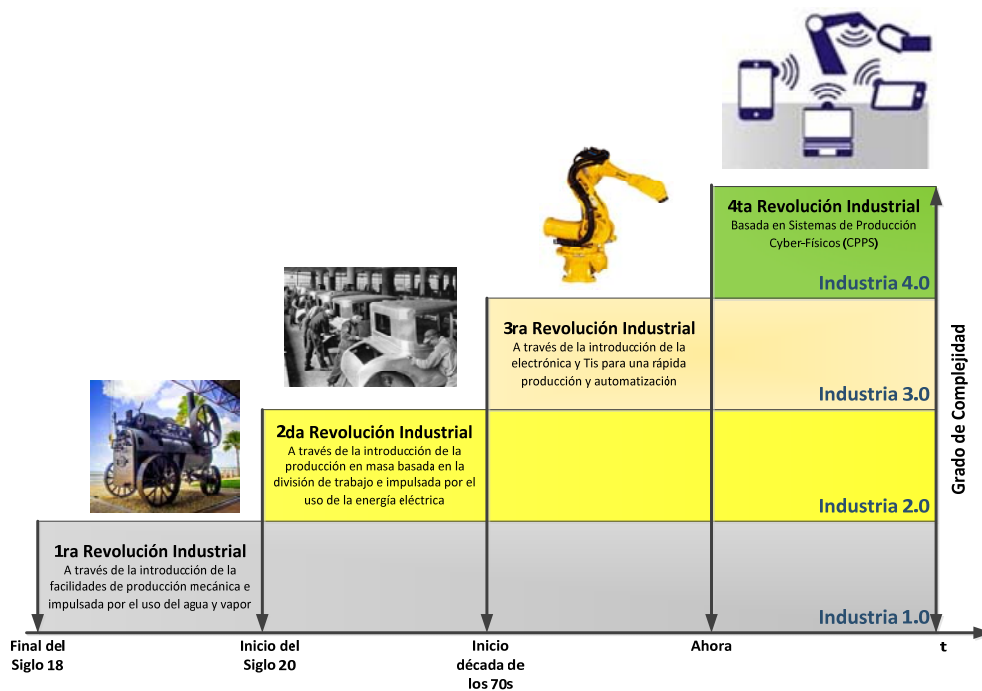


Figura 1: Historia de Evolución Industrial

trabajador y la máquina. El impacto de esta iniciativa es amplio, debido a que representa un punto de logro altamente dinámico, donde cada compañía, sea una gran fabricante de equipos, un proveedor de escala o un pequeño taller de trabajos, puede beneficiarse de las tecnologías y las plataformas de comunicación emergentes en el mercado de hoy actual. [15]

Sin embargo, para propósitos descritos en este capítulo, usaremos Industria 4.0 como una referencia más amplia, más general, a los drásticos cambios por venir en las fábricas del futuro. Vale la pena mencionar que el IoT Industrial y la Industria 4.0 tienen esencialmente una relación de causa y efecto. Esto significa que el IIoT es la base y dará lugar a la cuarta revolución industrial.

El término Industria 4.0 se ha referido a la cuarta revolución industrial (ver Figura 1), o la introducción de la tecnología de Internet en la industria manufacturera para hacer las fábricas más inteligentes, mejorando la ergonomía, la adaptabilidad y la eficiencia de los recursos, etc. [16]

Sin embargo, Industria 4.0 también es el término adoptado por una coalición de universidades, compañías, uniones de trabajadores y cuerpos de gobierno en Alemania. Esta iniciativa Industria 4.0 representa esa visión del país para el futuro de la manufactura, no sólo en Alemania, sino en todo el mundo. El propósito de esta iniciativa conjunta es mantener el liderazgo, tanto como usuario como desarrollador de tecnología de fábrica inteligente [17]. Esta iniciativa comprende una estructura conceptual detallada, una estrategia explícita con metas e hitos específicos, y fondos sustanciales de fuentes privadas y públicas. Es un modelo de una política convergente pública y privada para promover la fabricación avanzada.

Un enfoque clave de Industria 4.0 es equipar los futuros productos industriales y sistemas de producción con sistemas integrados de automatización inteligentes. Este objetivo se logrará desarrollando e implementando sensores, actuadores y controladores inteligentes que permitan la comunicación, el control de la operación, toma de decisiones y consumo de materias primas de manera eficiente. Industria 4.0 utiliza la capacidad de los CPPS para proporcionar comunicación e inteligencia para sistemas industriales. [18]

Industria 4.0 puede ofrecer las siguientes ventajas en los procesos de producción industrial: [19]

- **Mayor flexibilidad:** Las redes permiten estructurar los procesos de negocio de forma más dinámica. Los procedimientos de producción reaccionan con más flexibilidad a los cambios en la demanda.
- **Reducir los plazos de entrega:** La recopilación de datos sin interrupciones permite el uso rápido de datos e información relevantes para la producción para la toma de decisiones a corto plazo, y por lo tanto reduce los tiempos de espera para las innovaciones en el mercado.
- **Adaptación a las demandas de los clientes con tamaños de lote pequeños:** Industria 4.0 permite la incorporación de criterios específicos de cliente específicos en cuanto a planificación, configuración, pedido, diseño, producción y operación. También permite modificaciones en el tiempo.

- **Nuevas ofertas de servicios de downstream:** Industria 4.0 tiene el potencial para servicios de alto rendimiento para la evaluación a corto plazo de grandes datos.

Industria 4.0 depende principalmente de varias tecnologías innovadoras y se enumeran como sigue: [20]

- Tecnologías de la información y la comunicación para digitalizar la información e integrar diversos sistemas en todas las etapas del desarrollo y uso del producto.
- CPPS para monitorizar y controlar procesos y sistemas físicos. Se pueden utilizar sensores incorporados y robots inteligentes.
- Recopilación de grandes cantidades de datos, su análisis y explotación a través de la computación en nube y análisis de datos de gran volumen (Big Data).
- Interoperabilidad: Debido a que los CPS y los humanos están conectados a través del IoT y del IoS. Todos los CPS dentro de una planta son capaces de comunicarse entre sí a través de redes abiertas y descripciones semánticas.
- Virtualización: Los datos de los sensores están vinculados a modelos de simulación y modelos de plantas virtuales, creando una copia virtual del mundo físico.
- Capacidad en tiempo real, para que una planta pueda reaccionar a la falla de una máquina en el tiempo y reorientar un trabajo en proceso o productos a otra máquina.
- Orientación al servicio, los servicios de CPSs, seres humanos, y las compañías están disponibles sobre el IoS. Todos los CPS ofrecen sus funcionalidades como un servicio web encapsulado.

Las tecnologías de la Industria 4.0 usadas para el diseño e implementación de los diferentes componentes que intervienen en ella pueden ser resumidos en la Tabla 1.

Gracias a los conceptos de Industria 4.0, una fábrica inteligente puede ser definida como una fábrica donde los CPPS se comunican a través del IoT y ayudan a personas y máquinas en la ejecución de sus tareas [21]. La visión de Industria 4.0 va mucho más allá de la fabricación o de las primeras técnicas de fabricación integradas por computadora bajo el paradigma CIM. [22]

Tabla 1: Comparación de los principios para diseñar componentes en Industria 4.0

Principios	CPPS	IoT	IoS	Fábricas Inteligentes
Inter-operabilidad	X	X	X	X
Virtualización	X	---	---	X
Descentralización	X	---	---	X
Capacidad de Tiempo Real	----	---	---	X
Orientación al servicio	----	---	X	---
Modularidad	---	---	X	---

2.3 INTERNET DE LAS COSAS (IOT)

Internet de las Cosas (IoT) define una serie de tecnologías por las que objetos que tradicionalmente no han sido conectados ahora se van a conectar a la nube usando una serie de protocolos basados en el stack TCP/IP [23]. Estas tecnologías son los motores más importantes del crecimiento digital industrial. El concepto no es nuevo, el término fue acuñado por el pionero tecnológico británico Kevin Ashton en 1999 [24]. Sin embargo, en el pasado existían varios obstáculos que impidieron el despegue de IoT como: la seguridad en la comunicación, digitalización de las redes, etc.

El concepto de IoT en el mundo industrial no es tan radicalmente revolucionario como puede ser en la vida cotidiana. El desarrollo de Ethernet en la industria fue crucial para esta evolución debido a que permitió sistemas de supervisión cada vez basados más en la Nube (Cloud Automation) y disponibles en cualquier dispositivo móvil. Al fin y al cabo, el sensor siempre se ha comunicado con el PLC, lo que ha cambiado ha sido el medio, antes lo hacía mediante una señal analógica, ahora se lo hace mediante Ethernet o por algún mecanismo inalámbrico a través de la red de comunicación pública o privada.

En la actualidad, dichas limitaciones han ido desapareciendo y en otros casos disminuyendo, por lo que IoT ha ganado importancia de la mano de los avances en el desarrollo y utilización de hardware empotrado, la miniaturización de los dispositivos, el aumento en la capacidad de procesamiento y almacenamiento, la adopción de protocolos de comunicación más eficientes en la transmisión de datos, y de nuevos mecanismos/tecnologías de ahorro energético. En resumen, IoT está implementado por sistemas empotrados

o dispositivos hardware conectados a Internet con sensores integrados que pueden procesar (sin ser, ni convertirse en computadoras) y compartir información compleja sin intervención humana. [25]

2.3.1 Internet Industrial de las Cosas (IIoT)

Actualmente, las compañías de manufactura están implementando la conectividad inteligente de maquinaria y dispositivos en sus fábricas. Para distinguir estas aplicaciones del IoT de aquellas para consumidores y otros dominios, actualmente se usa el término Internet Industrial de las cosas (IIoT del inglés Industrial Internet of Things). [26]

El IIoT proporciona lo que se prevé será un cambio drástico que reconfigurará fundamentalmente la industria. IIoT reúne campos como la comunicación Máquina a Máquina (M2M del inglés Machine to Machine), el aprendizaje automático, computación en la Nube (Cloud Computing) y el análisis de Grandes Volúmenes de Datos (Big Data) para recopilar datos de máquinas y usarlos para ajustar las operaciones. [27]

IIoT representa un cambio del proceso de producción centralizado a un proceso de fabricación inteligente, gracias a los avances tecnológicos. Posee la capacidad de conectar cualquier dispositivo a una red de comunicaciones que permite recibir información desde muchas fuentes para ser almacenada, transferida, analizada, personalizada o automatizada sin intervención humana. Los problemas pueden ser resueltos y las decisiones pueden ser tomadas a partir del análisis de datos, manteniendo una calidad constante y reduciendo el riesgo de error humano. Tanto la cadena de producción como la cadena de valor se verán afectadas y, a través de ellas, los productos y servicios se beneficiarán del IIoT.

El IIoT impulsará la competitividad, la eficiencia y garantizará el desarrollo gracias a la tecnología. No existe una aplicación concreta que pueda denominarse IIoT, sino que representa un conjunto de tecnologías que gradualmente permitirán, concretar la visión de los objetos cotidianos a nivel industrial conectados a Internet [27]. Para una amplia introducción de IIoT a nivel industrial, se tendrán que ir solucionando problemas como: la seguridad, la adopción de nuevos mecanismos y tecnologías de redes de comunicación

especializadas y de amplia difusión, los lenguajes de programación que mejor comprendan la comunicación entre los elementos físicos para obtener aplicaciones que sean construidas pensando en usuarios que esperan resultados inmediatos en sus dispositivos móviles en cualquier parte del mundo.

La visión de IIoT solo podrá llevarse a cabo si la comunicación entre los componentes principales del sistema se basa en un estándar global que pueda satisfacer los complejos requisitos exigidos. Además de utilizar un modelo productor/consumidor (muy eficiente en el uso de recursos de red y que permite el intercambio de información “*de uno a varios*”) se requiere un modelo de comunicación cliente/servidor orientado al establecimiento de conexiones seguras, con el que sea posible gestionar el intercambio bidireccional de información que permite enviar las órdenes de control a los actores. [28]

Para garantizar el uso adecuado de la información a nivel industrial de acuerdo a los conceptos de IIoT, ésta debe ir acompañada de un modelo semántico que describa los tipos de datos y su propósito [29]. La agregación de información en múltiples capas permite la incorporación de metadatos adicionales y por ello es de vital importancia utilizar estándares que sean escalables y dispongan de capacidad para integrar la información a través de todas las capas, además de ser independientes del vendedor y de la plataforma.

IoT en general requiere 3 capas, de acuerdo con la AIIoTI (Alliance for Internet of Things Innovation): [30]

- **La capa de aplicación**, la cual contiene las comunicaciones y los métodos de interfaz utilizadas en las comunicaciones de proceso a proceso.
- **La capa del IoT** agrupa funciones específicas, tales como el almacenamiento e intercambio de datos, y poniéndolas disponibles a la capa de aplicación a través de interfaces, comúnmente conocidas como interfaces de programación de aplicaciones (API). La capa del IoT hace uso de los servicios de la capa de red.
- **La capa de red** provee servicios que se pueden agrupar en *servicios de datos*, proporcionando conectividad de corto y largo alcance, así como la transmisión de

bio, sea posible controlar y supervisar de forma automática procesos en los que intervienen las mismas. Por tanto, el foco principal de aplicación de M2M se ubica en los entornos relacionados con la telemetría y/o el telecontrol industrial.

Los modelos de negocio que surgen de estas nuevas posibilidades de comunicación están ligadas principalmente al ámbito de la logística y el mantenimiento particularmente en aplicaciones de monitorización de la condición y mantenimiento preventivo. Otro motivo para conectar activos y dispositivos usando estas soluciones ha sido mejorar su eficiencia operacional. El gasto se justifica por la promesa de la reducción de paradas no programadas, uso de supervisión remota para identificar fallos de equipos o predecir fallos que luego se concretan más exactamente. [32]

En el ámbito de M2M están surgiendo redes muy complejas de sistemas inteligentes. Su objetivo no solo es que máquinas y dispositivos de campo se conecten a las redes de comunicación para el intercambio de datos, sino que también pueden, gracias al incremento que se está produciendo en su capacidad de computación, procesarlos y combinarlos con información procedente de otros elementos de la planta. De este modo, los sistemas podrán consumir y producir nuevos tipos de información agregada que aporten un valor añadido para el usuario final, proporcionar una estrategia de mantenimiento para el personal técnico o enviar información sobre su histórico de mantenimiento en lugar de ceñirse únicamente a intercambiar datos como temperatura, presión del aceite, etc.

2.5 SISTEMAS CIBER-FÍSICOS (CPS)

En los últimos años, el término Sistemas Ciber-Físicos (CPS del inglés Cyber-Physical Systems) ha sido usado como un sinónimo tanto en la investigación como en la industria, siendo este concepto acuñado por Lee [33]. El término Ciber indica computación, comunicación y control de sistemas discretos, conmutados y lógicos; el término físico se refiere a los sistemas naturales y humanos gobernados por las leyes de la física y operados en tiempo continuo o discreto. [34]

Se puede decir en consecuencia, que un CPS es un sistema formado por un conjunto de dispositivos que colaboran mutuamente y está diseñado para controlar objetos físicos (mecánicos, electrónicos, etc). La comunicación se lleva a cabo a través de una infraestructura de datos como Internet, computación en la nube o sistemas industriales. Los sistemas empotrados tradicionales pueden considerarse como casos especiales de CPS autónomos. [35]

Un CPS generalmente consta de dos componentes funcionales principales:

- La conectividad avanzada que asegura la adquisición de datos en tiempo real desde el mundo físico y la retroalimentación de la información desde el ciberespacio;
- La gestión inteligente de datos, computacional y la capacidad de análisis que construye el espacio cibernético. [36]

El paradigma del IIoT describe las redes distribuidas, que a su vez están compuestas de redes de objetos inteligentes. La característica principal de una CPS es que las tecnologías de información y comunicación son desarrolladas y dirigidas hacia la creación de sistemas virtuales con capacidades equivalentes a componentes físicos. Los CPS enlazan datos del mundo real a través de canales de comunicación; de esta manera se puede utilizar la potencia de cálculo distribuida para resolver problemas dentro de una red. El IoT y el CPS no son conceptos fundamentalmente nuevos. De hecho, Simon [37] ya identificó la importancia y los beneficios de combinar los dominios tanto físicos como virtuales. Su enfoque fue presentado hace muchos años, cuando no todas las plataformas empotradas y técnicas de fabricación se encuentran tan desarrolladas como actualmente.

Los enfoques tecnológicos de Internet que ayudan al concepto de CPS comprenden los siguientes conceptos:

- Internet de Servicios (IoS): Comprende nuevos paradigmas de servicios como el proporcionado por arquitecturas orientadas a servicios (SOA) o la tecnología REST.

- Internet de Datos (IoD): Permite almacenar y transferir datos en masa de forma adecuada, y proporcionar nuevos e innovadores métodos de análisis para interpretar éstos datos.

J. Lee [38] propone una estructura CPS de 5 niveles en la cual se define cómo las personas construyen un CPS desde la adquisición inicial de datos, luego el análisis y hasta la creación final de valor. Un resumen de esta arquitectura se describe en la Tabla 2. Entre los niveles de esta arquitectura, los niveles de cognición y configuración son los más difíciles de lograr.

Tabla 2: Arquitectura 5C para la implementación de CPPS

Arquitectura	Conceptos
Conexión inteligente	La adquisición de datos precisos y confiables de las máquinas y sus componentes es el primer paso
Datos a información	Deducción de información significativa a partir de los datos.
Ciber	Este nivel actúa como un centro de información central en esta arquitectura. Al reunir información masiva, se deben realizar análisis específicos para extraer información adicional para proporcionar una mejor visión.
Cognición	La implementación de CPS a este nivel genera un conocimiento profundo del sistema monitorizado. Una correcta presentación de los conocimientos adquiridos a los usuarios expertos apoya una decisión correcta.
Configuración	Este nivel es la retroalimentación del espacio cibernético al espacio físico y actúa como un control de supervisión para hacer que las máquinas se autoconfiguren y se adapten a sí mismas. Esta etapa actúa como un Sistema de Control de Resiliencia (RCS) para aplicar decisiones correctivas y preventivas.

2.5.1 Sistemas de Producción Ciber-Físicos (CPPS)

La aplicación de CPS en sistemas de producción conduce a los llamados Sistemas de Producción Ciber-Físicos (CPPS del inglés Cyber-Physical Production Systems), en los cuales productos, máquinas y otros recursos están representados por CPS los cuales permiten que se compartan información y servicios en toda la jerarquía de la industria dando un valor añadido a la red de comunicaciones industriales y a la red de gestión y administración.

Con el diseño de nuevos sensores y actuadores inteligentes, los CPPS poseen actualmente la capacidad de recoger directamente datos del proceso y evaluarlos en tiempo real. De esta manera se consigue una conexión directa entre el mundo físico y el mundo

digital [39]. Los datos del proceso se ponen a disposición de distintos servicios conectados en red, que los utilizan para enviar órdenes a los actuadores, los cuales ejecutan acciones en el mundo físico.

Un CPPS está basado en una lógica de producción descentralizada y que posee la capacidad de mantener comunicaciones usando diferentes protocolos industriales y de TI. Un CPPS ofrece ventajas en términos de transparencia, adaptabilidad, eficiencia de recursos y versatilidad sobre los sistemas de producción tradicionales. Es decir, los CPPS son unidades que tienen acceso casi instantáneo a información relevante y parametrización de máquinas, procesos de producción y al propio producto fabricado. [40]

Los CPPS se pueden caracterizar por las siguientes características: [41]

- Una parte física, por ejemplo, sensores y actuadores capturan datos físicos directamente. Esto permite una influencia directa en los procesos físicos.
- Una parte de comunicación, por ejemplo, conectada a redes digitales: inalámbrica, enlazada, local, global. Esto permite el uso de datos y servicios disponibles globalmente.
- Una parte de cálculo, por ejemplo, guardar y evaluar datos e interactuar sobre esta base, activa o reactiva con mundos físicos y digitales.
- Una capa de interacción para HMI, por ejemplo, presenta una gama de interfaces para la interacción hombre-máquina multimodal. Esto proporciona facilidades dedicadas para la comunicación y el control, como el control por el habla y los gestos.

Por lo tanto, podemos decir que los CPPS consisten en elementos y subsistemas autónomos y cooperativos que se conectan entre sí, en todos los niveles de producción, desde los procesos hasta las redes de producción y logística.

En la fabricación, los CPPS pueden mejorar la calidad y la productividad a través de pronósticos inteligentes y diagnósticos utilizando gran volumen de datos dados por

diferentes máquinas, sensores en red y sistemas. Los CPPS pueden modelar su funcionamiento y prever su comportamiento emergente plantea una serie de tareas de investigación básicas y orientadas a la aplicación. [42]

Lo fundamental en las nuevas investigaciones en el caso de los CPPS es explorar las relaciones de autonomía, cooperación, optimización y capacidad de respuesta. Se puede indicar que la integración de los enfoques analíticos y basados en la simulación de procesos será más significativa que nunca. Se debe afrontar los desafíos de operar redes de sensores, manejar grandes volúmenes de datos, así como problemas en la recuperación de información, representación e interpretación, con especial énfasis en aspectos de seguridad. Los CPPS permitirán y apoyarán la comunicación entre humanos, máquinas y productos por igual.

El concepto de los CPPS rompe parcialmente con la tradicional pirámide de automatización [40] como se puede observar en la Figura 3. Actualmente sigue existiendo los típicos niveles de control y campo los cuales incluyen PLCs cercanos a los procesos técnicos para poder ofrecer el mejor rendimiento para los bucles de control críticos; mientras que, en los niveles más altos de la jerarquía de la pirámide, se tiene una característica más descentralizada de funcionamiento y es aquí donde fundamentalmente se utilizarán los CPPS.

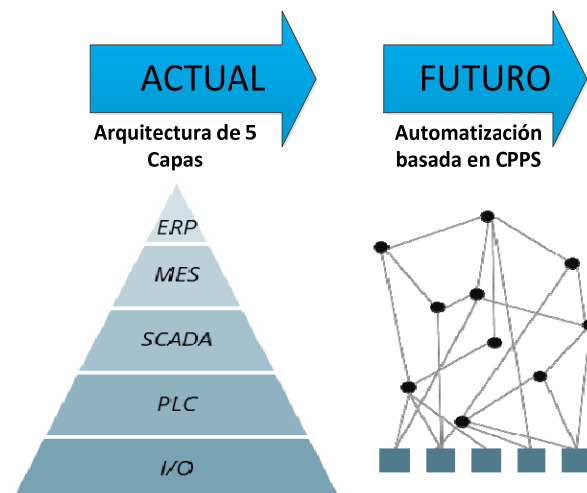


Figura 3: Arquitectura de Automatización Industrial

Un problema importante es la seguridad en el intercambio de información de los sistemas CPPS. Se necesitan tecnologías basadas en SOA o TI para proporcionar una infraestructura segura y para gestionar activos de información en el entorno de fabricación industrial. Los CPPS pueden ser aplicables para la gestión de grandes datos y aprovechar la interconectividad de las máquinas para alcanzar la meta de las máquinas resistentes, inteligentes y auto-adaptables [43]. El papel del análisis de grandes volúmenes de datos para CPPS llegará al diseño, fabricación, mantenimiento, uso y reutilización cuando los procesos manejen nuevos tipos de datos y problemas en la fabricación en las llamadas Fábricas Inteligentes. [44]

2.6 TECNOLOGÍAS DE COMUNICACIÓN EN LA AUTOMATIZACIÓN INDUSTRIAL

En este apartado se verán las tecnologías más usadas para la comunicación industrial, debido a que es de suma importancia identificar la tecnología adecuada para su implementación en CPPS.

2.6.1 Arquitectura Orientada a Servicios (SOA)

Existen muchas definiciones de SOA, variando de acuerdo a la formación del autor y dominio de aplicación. Puesto que esta tesis doctoral se enfoca en los dispositivos empujados usados a nivel de planta de una industria manufacturera para diseñar plataformas CPPS, el autor se adhiere a la definición presentada en la publicación pionera de Jammes y Smith [45]: “*SOA es un conjunto de principios para construir sistemas autónomos y a la vez interoperables*”. Aunque simple y concisa esta definición sin duda incompleta, trata de demostrar que SOA promueve una fluida combinación entre autonomía e interoperabilidad, conceptos contradictorios por definición.

Un servicio se considera autónomo puesto que se crea y opera independientemente de su entorno y es autocontenido. A la vez es interoperable a través de una interfaz que expone su funcionalidad a dicho entorno, abstrayendo por tanto los detalles de su implementación. Sin embargo, SOA no es sólo una arquitectura sino también, un modelo de

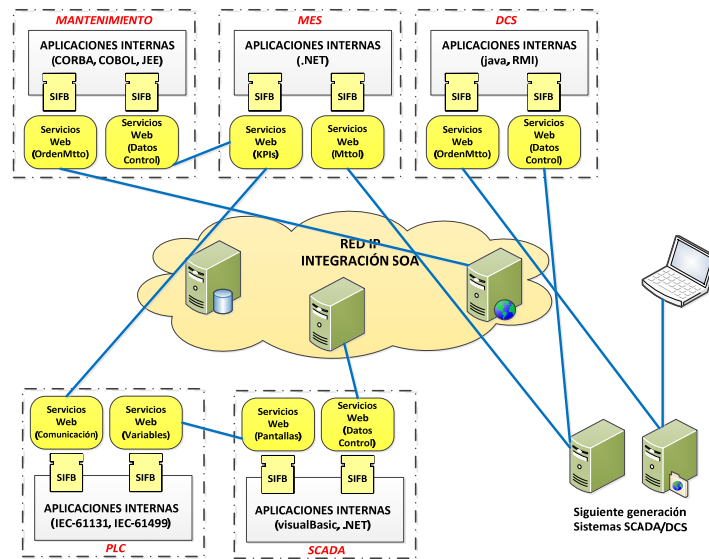


Figura 4: Automatización Industrial en ambiente SOA

programación que define una nueva forma de pensar acerca de la implementación de sistemas software. [46]

SOA por definición es agnóstica con respecto a plataformas, lenguajes e implementaciones; justamente el uso de estándares abiertos, permite implementar SOA de forma neutral con respecto a tecnología. Esta característica hace a SOA particularmente aplicable a entornos heterogéneos donde la interoperabilidad sea esencial. Los servicios interactúan entre sí a través de mensajes, con un esquema que define su formato, un contrato para definir sus intercambios y la definición de una política de cómo deben ser intercambiados. SOA permite la aproximación al diseño de aplicaciones complejas basada en: la identificación de los servicios que ofrecería, la definición de esos servicios, la organización de las interacciones entre esos servicios, la importancia y descripción rigurosa de las interfaces, el tratamiento automático para generar código de implementación, como podemos observar en la Figura 4.

Según Milagaia [47] la tecnología existente no proporciona una solución completa, independiente de la plataforma y formato de datos. El autor postula que debe existir un resultado intermedio, en el cual coexistan sistemas preexistentes junto con aquellos que

utilizan los nuevos arquetipos tecnológicos. A tal fin, paradigmas tales como los emergentes WS* y SOA, ambos soportados por distintas plataformas independientemente de la tecnología utilizada, parecen tener muchas posibilidades.

Shen et al. [48] señalan como otro tema difícil, aún no resuelto, la integración completa de control y planificación de procesos de fabricación, particularmente porque está en juego la información en tiempo real obtenida por sistemas de colección de datos, problema nada trivial. Debe establecerse el mapeo correcto para conectar los bits de información de distintos sistemas con diferentes identificadores, representaciones de datos y niveles de granularidad: desde segundos (o en algunos casos milisegundos) a meses y de equipamientos aislados al manejo de puntos de producción completos.

La integración de dispositivos a nivel de planta basada en SOA, ya fue realizada a nivel de prototipo dentro del alcance del proyecto SOCRADES y, está en etapa de realización de pruebas para su evaluación en un ambiente real de fabricación [49]. Sin embargo, la implementación de WS* en ellos tiene ciertos requisitos en cuanto a capacidad de memoria, computacional y de almacenamiento que no son desdeñables. Este hecho constituye una barrera en el corto plazo, en el caso de dispositivos con tecnologías más viejas y con ciclos de vida en algunos casos de décadas, como los que aún suelen encontrarse en la mayoría de plantas manufactureras. La transición hacia un nivel de planta totalmente basado en SOA se realiza mediante pasarelas inteligentes, llamados gateway o mediadores de servicios en el proyecto SOCRADES. Sin embargo, la integración a tan bajo nivel es muy costosa e implicaría rediseñar o introducir cambios significativos en los dispositivos de control industrial.

2.6.2 Servicios WEB

Un Servicio Web (WS* del inglés Web Services) es un componente software que puede ser registrado, descubierto e invocado mediante protocolos estándares de Internet. Utiliza un conjunto de protocolos y estándares que permiten el intercambio de datos entre aplicaciones heterogéneas.

Cada WS* se responsabiliza de realizar un conjunto de funciones concretas y bien definidas que actúan como componentes independientes y que se pueden integrar para formar sistemas distribuidos complejos tanto a nivel empresarial y, últimamente, a nivel de sistemas de automatización. Un servicio es una función bien definida, autocontenida (es decir, siempre proporciona la misma funcionalidad independientemente de los otros servicios) y, que no depende de contexto o estado de otros servicios [46]. Los servicios pueden combinarse dinámicamente en tiempo de ejecución, para constituir aplicaciones compuestas ya sea, ensamblando aquéllos que se ejecutan encima de aplicaciones pre-existentes u otros creados como parte de nuevas aplicaciones.

La Definición según la World Wide Web Consortium (W3C), nos dice que “*Un Servicio Web es una aplicación software identificada por un URI (Uniform Resource Identifier), cuyas interfaces se pueden definir, describir y descubrir mediante documentos XML. Los Servicios Web hacen posible la interacción entre agentes software (aplicaciones) utilizando mensajes XML intercambiados mediante protocolos de Internet.*” [50]

Las características principales de los servicios Web, son las siguientes:

- **La utilización de estándares de internet:** La única forma para que los servicios Web sean utilizados por la cantidad de sistemas heterogéneos existentes en Internet y a nivel industrial es el empleo del protocolo de transferencia de datos HTTP utilizado por todos los navegadores Web y XML.
- **Son basados en tecnologías de paso de mensaje:** La interacción entre el cliente y el proveedor del servicio es empaquetada en unidades autodescriptivas denominadas mensajes. Dicha interacción se describe en función de los mensajes intercambiados.
- **Combinan lo mejor de la tecnología de componentes y de la tecnología Web:** Los servicios Web presentan una funcionalidad de caja negra que puede ser reutilizada sin preocuparse de cómo es implementada y ello proporciona interfaces bien definidas.

En la automatización industrial contrariamente al principio de la era CIM, hoy día existen componentes tecnológicos básicos muy potentes, pero aún es evidente la necesidad de capas abstractas de interoperabilidad. El problema principal no ha cambiado mucho a través de los años: encontrar interfaces apropiadas entre gran cantidad de aplicaciones que en su mayoría son no interoperables. La adopción de WS* como columna vertebral de comunicación en la empresa, permitiría eliminar el altamente costoso software propietario. [51]

2.6.2.1 SOAP

Simple Object Access Protocol (SOAP) es una especificación de WS* que básicamente describe un formato de mensajes para comunicar aplicaciones. La filosofía de SOAP, para lograr dicha comunicación, es no inventar una nueva tecnología, sino combinar tecnologías existentes y de amplia aceptación en la industria de software. En particular, combina XML para la codificación de los mensajes y HTTP como protocolo de transporte (aunque no se excluye el uso de otros protocolos de transporte) [52]. En definitiva, SOAP define un mecanismo para el intercambio de información, estructurada y tipeada, entre pares de aplicaciones en un entorno distribuido, teniendo como objetivos de diseño la simplicidad y la extensibilidad.

SOAP no define por sí mismo la semántica de las aplicaciones, como lo haría un modelo de programación o algún tipo de semántica específica de una implementación, sino que proporciona un mecanismo simple para expresar la semántica de las aplicaciones, mediante un modelo modular de empaquetado de mensajes y la definición de como codificar los datos de las aplicaciones en dichos módulos.

Es posible ver a SOAP desde distintos puntos de vista: [53]

- Como un mecanismo para invocar métodos en servidores, servicios, o componentes, para lo cual se define en la especificación una metodología para encapsular e intercambiar invocaciones RPC, en los mensajes, usando la extensibilidad y flexibilidad que proporciona XML.
- Como un protocolo para intercambio de mensajes (sincrónicos o asincrónicos).

- Como un formato para intercambio de documentos XML.

2.6.2.2 RESTful

La Transferencia de Estado Representacional (REST acrónimo del vocablo inglés Representational State Transfer) es una arquitectura propuesta por Roy Fielding en su disertación de doctorado [54]. El objetivo principal de REST es proporcionar interoperabilidad entre sistemas cliente – servidor que operen en la web. REST se basa en la existencia de recursos que son elementos de información, y como éstos deben ser representados y el uso de enlaces entre los diferentes recursos para permitir la navegación entre ellos. Para manipular estos recursos, los componentes de la red que pueden ser clientes o servidores se comunican a través de una interfaz estándar como es el protocolo HTTP e intercambian representaciones de estos recursos.

En REST, la comunicación entre cliente - servidor es sin estado y cuando un cliente solicita un recurso, éste es direccionado de manera única usando un URI. El servidor no tiene ninguna información sobre los clientes (no hay diferencia entre dos solicitudes del mismo cliente), cada mensaje HTTP contiene toda la información necesaria para comprender la petición. El recurso solicitado se entrega al cliente utilizando un modelo para representarlo. Sin embargo, en la práctica, muchas aplicaciones basadas en HTTP utilizan cookies y otros mecanismos para mantener el estado de la sesión.

Un servicio web basado en REST se denomina servicio web RESTful y aprovecha el uso de métodos HTTP para facilitar la operación CRUD (crear, leer, actualizar, eliminar) en los recursos del sistema. Específicamente las operaciones CRUD se clasifican en: POST (utilizado para la operación de creación), GET (utilizado para la operación de lectura), PUT (utilizado para actualizar la operación), DELETE (utilizado para la operación de eliminación).

Los servicios web RESTful se basan en una comunicación sin estado entre el cliente y el servidor. Tal enfoque necesita que cada solicitud sea independiente de cualquier contexto almacenado en el servidor, y cada una de ellas debe contener toda la información

necesaria para realizar el servicio correctamente: incluye todos los parámetros y el contexto (dentro del encabezado y el cuerpo de una solicitud HTTP), necesario para generar una respuesta por parte del servidor.

2.6.3 OPC-UA

Las más modernas herramientas de supervisión y gestión de procesos de fabricación, precisan recopilar en tiempo real información procedente de medios productivos cuyo funcionamiento está gobernado, en muchos casos, por controladores de diferentes fabricantes. Las investigaciones típicas de integración vertical usualmente separan los sistemas de automatización en diferentes capas. La estructura resultante es usualmente conocida como la pirámide de automatización. Aunque este es un enfoque fiable, esto significa que todas las peticiones de los sistemas como, por ejemplo: invocación de funciones, adquisición de datos, transferencia de datos, etc; deben ser transferidos de una capa a otra, y al realizar este intercambio generalmente pasan por interfaces de comunicación propietarias en donde deben realizarse las traducciones asociadas, pero a veces, estas traducciones son propensas a errores. [55]

A pesar de la existencia de estándares de comunicación como OPC-DA (acrónimo de Ole for Process Control-Data Access) y de programación como la norma IEC 61131-3, la captura automática de datos en planta suele ser, todavía en nuestros días, un apartado extremadamente complicado y problemático. Las causas de dicho problema radican en las diferentes estructuras, contenidos y vistas de los datos que residen en cada controlador y para las que no existe, todavía, ninguna norma general.

La Arquitectura Unificada UA (acrónimo del vocablo inglés Unified Architecture), propuesta por OPC Foundation, es capaz de responder a las preguntas de los problemas que actualmente tiene la integración de datos a nivel de planta sobre cómo se localiza al interlocutor de una comunicación, cómo se identifican los datos, metadatos y metadatos de la lógica de control y cómo se define la información que debe hacerse visible a otros dispositivos. Sin embargo, todos los puntos de vista de la problemática de integración de

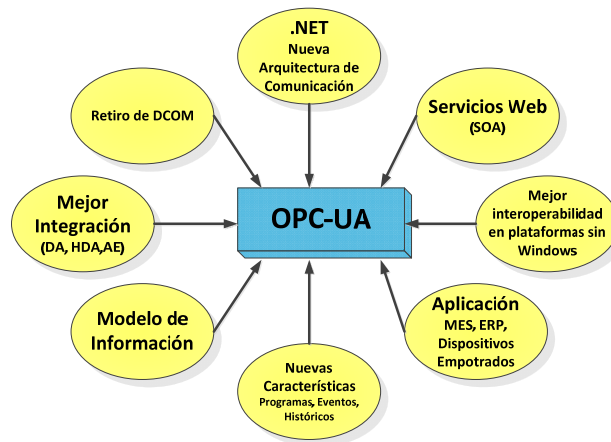


Figura 5: Tecnología de OPC-UA

datos a nivel de planta deben ser tenidos en cuenta si se desea que los diferentes fabricantes puedan intercambiar información entre ellos de forma adecuada.[56]. OPC-UA proporciona un marco de interoperabilidad industrial del futuro.

OPC-UA tiene que satisfacer y mejorar los requisitos no funcionales de la OPC clásica y proporcionar, por ejemplo, una comunicación robusta, fiable, de gran rendimiento, adecuada para la automatización entorno al paradigma de la Industria 4.0, como se puede observar en la Figura 5. OPC-UA reúne las distintas especificaciones de OPC clásica y ofrece un punto único de acceso a un sistema industrial. Está diseñado para acceder a una gran cantidad de datos de dispositivos en tiempo real usando la infraestructura de red estándar, manteniendo un alto rendimiento.

A diferencia de la OPC clásica, OPC-UA proporciona un conjunto reducido y único de servicios genéricos para el acceso a toda la información [57]. Una de las mayores ventajas del modelo de información de OPC-UA se pone de manifiesto en las aplicaciones que serán clientes de OPC-UA a nivel industrial. Las aplicaciones podrán consultar, de forma segura con respecto al tipo de datos, el espacio de direcciones de todos los controladores de la planta y también obtener información sobre los procesos que tienen en ejecución y sus tiempos de ciclo [58]. De esta manera la pirámide de automatización queda truncada ya que esta no necesita pasar por interfaces propietarias para ingresar a sistemas MES y ERP, como se observa en la Figura 6.

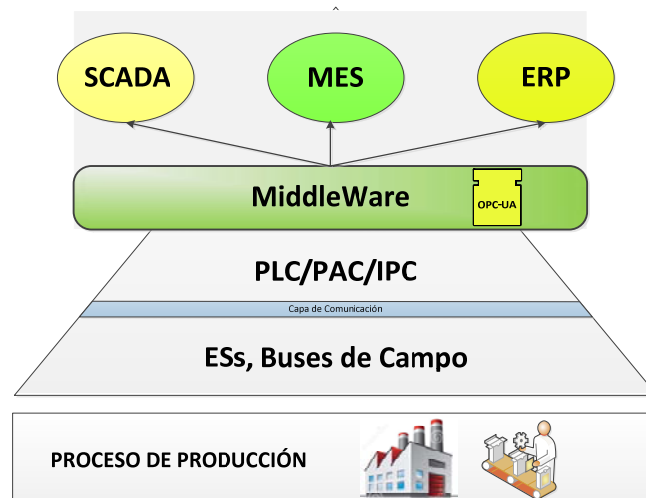


Figura 6: Arquitectura SOA en Automatización

El objetivo principal de la OPC-UA es mantener la funcionalidad de la OPC clásica y pasar de la tecnología COM/DCOM de Microsoft a una tecnología de servicios de vanguardia. Utilizando la tecnología de WS*, OPC-UA es independiente de la plataforma y, por lo tanto, puede aplicarse en situaciones en las que la OPC clásica ya no se utiliza [59]. OPC-UA se puede integrar sin fisuras en los sistemas industriales y funciona no sólo en sistemas Unix/Linux, Java, Windows, etc., sino también en los controladores y dispositivos inteligentes que tienen sistemas de funcionamiento específicos con capacidad de procesamiento en tiempo real.

Por supuesto, la compatibilidad con las especificaciones OPC anteriores fue un requisito para OPC-UA. OPC-UA se puede adaptar a distintos tamaños de hardware. Es posible utilizar aplicaciones de OPC-UA en dispositivos empotrados con recursos de hardware muy limitados, así como en máquinas muy potentes. Normalmente, servidores que funcionan en entornos tan diferentes no proporcionan la misma información. Es poco probable que el servidor del dispositivo empotrado proporcione un largo historial de datos, y sólo soportará unos pocos clientes, mientras que otros servidores pueden proporcionar varios años válidos de datos históricos y soportar miles de clientes. Esto depende de las aplicaciones a nivel industrial.

OPC-UA define dos pilares principales que sustentan la interoperabilidad: la infraestructura de comunicaciones y el metamodelo, como se observa en la Figura 7. La infraestructura de comunicaciones define cómo se intercambia la información, y el metamodelo define qué información se debe intercambiar. La infraestructura de comunicaciones por WS* puede facilitar su integración con aplicaciones empresariales y el acceso a Internet y, por otra, el uso de un protocolo de transmisión binario basado en TCP permite un intercambio eficiente de datos en términos de prestaciones [57]. Además de todo ello, se integra una serie de mecanismos especiales definidos en el API y en los protocolos que garantizan una comunicación fiable en el caso de trabajar con sistemas distribuidos.

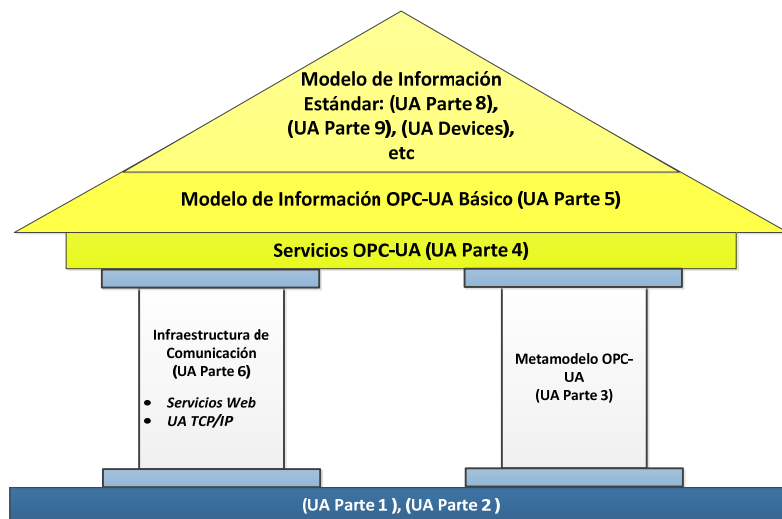


Figura 7: Pilares OPC-UA

Como se explica anteriormente, la arquitectura OPC-UA permite también suministrar el modelo de información, esto significa que la meta información se proporciona junto con los datos. Los aspectos del modelado de la información de OPC-UA también son escalables.

Un servidor puede proporcionar cualquier tipo de información, desde un modelo muy sencillo, similar a OPC clásica, hasta modelos muy complejos que proporcionan metadatos muy sofisticados sobre los datos suministrados. Un cliente puede no tener en cuenta esta información complementaria y proporcionar una visión simple de los datos o utilizar los metadatos proporcionados por el servidor. [60]

OPC-UA no es compatible con OPC clásica, ya que utiliza una tecnología diferente para la transmisión de datos. Para cumplir este requisito, sin embargo, la Fundación OPC suministra wrappers y pasarelas que, o bien adaptan de forma automática los servidores existentes a los clientes OPC-UA, o proporcionan un servidor proxy a los clientes de OPC clásica para acceder al servidor OPC UA.

La norma OPC-UA consta de 13 partes diferentes de los cuales algunos ya han sido adoptados por la Comisión Electrotécnica Internacional como el estándar IEC 62541. En pocas palabras, esta norma posee diferentes capítulos los cuales resumimos a continuación: el primer capítulo nos indica un panorama general de esta norma, un concepto amplio de seguridad se presenta en el capítulo 2, el modelo del Espacio de Direcciones (vocablo que viene de la palabra inglesa Address Space) se da en el capítulo 3, en el capítulo 4 se dan sus conceptos y funciones servicios abstractos, el capítulo 5 nos indica el modelo de información, los mapeos de las tecnologías en el capítulo 6, los perfiles en el capítulo 7, el acceso a los datos se da en el capítulo 8, la visualización de datos en el capítulo 9, el método de las llamadas y el acceso histórico de datos en el capítulo 10 y 11, respectivamente y funcionalidades para servidores OPC-UA en los capítulos 12 y 13. Esto se expresa gráficamente en la Figura 8.



Figura 8: Partes de OPC-UA

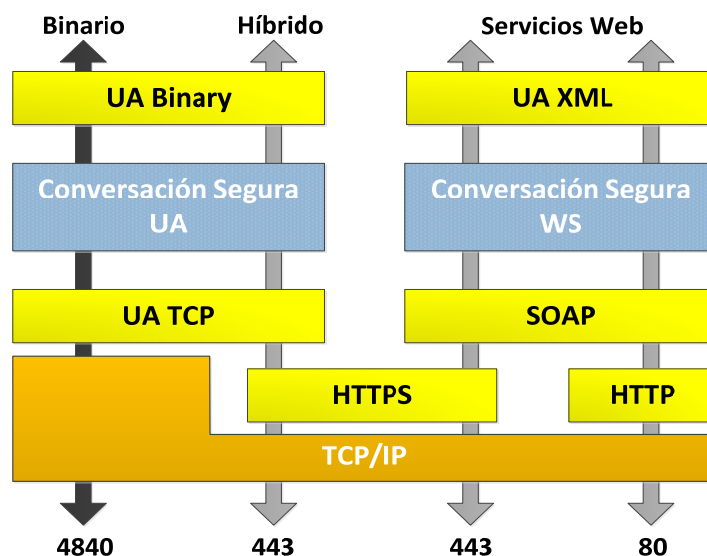


Figura 9: Perfiles de Transporte de OPC-UA

En las siguientes secciones se dará a conocer ciertos conceptos básicos con respecto a tres aspectos importantes de OPC-UA como son: el modelo de información, los servicios de comunicación y la seguridad.

2.6.3.1 Servicios de Comunicación

OPC-UA utiliza una arquitectura cliente – servidor siguiendo el paradigma de la Arquitectura Orientada a Servicios (SOA). La capa de la arquitectura de la comunicación está basada en el mapeo de dos diferentes tecnologías que son WS* y el protocolo en formato binario. Adicionalmente usa reglas de modelado, como vimos anteriormente, con el objetivo de diseñar y modelar los datos del servidor o cliente. Se ocupa de los servicios básicos genéricos utilizados para la comunicación entre servidores y clientes.

Con el fin de hacer que los servicios abstractos de OPC-UA sean aplicables al reunirse los requisitos específicos, se han especificado diferentes mapeos de tecnología de comunicación para codificaciones y protocolos de transporte. La Figura 9 ofrece una visión general sobre las asignaciones definidas también teniendo en cuenta las soluciones de seguridad. Para ofrecer comunicaciones fiables, OPC-UA ha incorporado mecanismos

capaces de afrontar problemas, como los mensajes perdidos. Dispone de seguridad incorporada, un requisito que se ha vuelto cada vez más importante en entornos donde desde la red de la oficina se debe acceder a los datos de la planta.

OPC-UA define dos modelos de codificación de datos:

- **UA Binario:** Esta especificación define un mensaje de encriptación binario no portable, optimizado para cualquier porte y tipo de mensajes y para una rápida codificación y decodificación. La especificación descansa en tipos de datos primitivos para los cuales la codificación binaria está definida. La codificación excluye los tipos y el campo de nombre de información, porque las aplicaciones esperan tener un avanzado conocimiento de los servicios y de la estructura de los datos a ser transmitidos.
- **UA XML:** La norma también define una representación de los elementos en el modelo de objetos en formato XML el cual es muy útil para su uso en WS* SOAP/HTTP. La codificación que usa UA XML es la definida por el esquema W3C XML.

Adicional han sido definidos dos mapeos a nivel de transporte:

- **UA TCP (Nativo):** Se trata de un protocolo específico de OPC-UA basado en la tecnología TCP, de esta manera se establece un canal full-duplex para transmitir datos binarios entre clientes y servidores OPC-UA. Este TCP nativo está diseñado para mejorar los errores de las versiones TCP anteriores ya que permite trabajar en un canal seguro el cual está implementado en una jerarquía superior. Esta opción cumple con los requisitos de comunicación para sistemas de control.
- **SOAP/HTTP (Servicios Web):** En este mapeo los mensajes OPC-UA son serializados en formato XML, para luego ser precedidos por cabeceras SOAP y permitir el intercambio usando el modelo de pregunta-respuesta, definido SOAP. Adicionalmente, los protocolos HTTP y HTTPS son usados para estos intercambios. El mapeo SOAP permite la comunicación a través de Internet, por ejemplo, para aplicaciones empresariales.

2.6.3.2 Servicios Base

OPC-UA define los servicios necesarios para navegar a través del espacio de direcciones, leer o escribir en las variables, o la suscripción en las modificaciones de datos y eventos. Los servicios de OPC-UA están organizados en conjuntos lógicos, los llamados conjuntos de servicios. La solicitud de servicio y la respuesta se transmiten a través del intercambio de mensajes entre clientes y servidores. Los mensajes OPC-UA como se ha dicho se intercambian ya sea a través de un protocolo binario-OPC dado sobre TCP/IP o como un WS*. Las solicitudes se suelen dar por estos dos tipos de protocolo, por lo que el operador del sistema puede elegir la mejor opción [2]. OPC-UA ofrece un total de 9 conjuntos de servicios básicos (ver Figura 10).

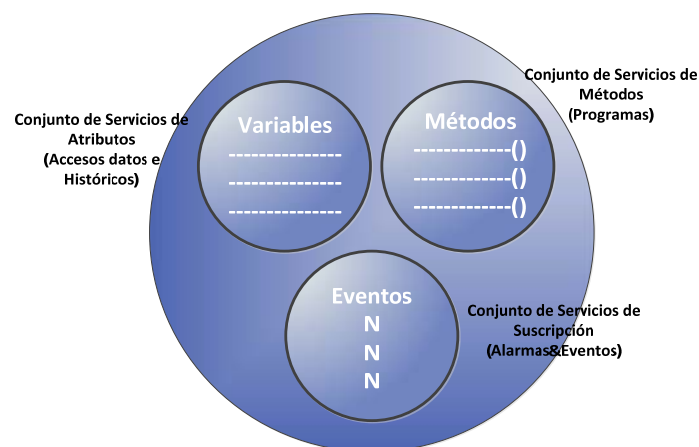


Figura 10: Conjuntos de Servicios de OPC-UA

Los perfiles permiten especificar un subconjunto de todos los servicios que soporta un servidor. Los conjuntos individuales se describen brevemente a continuación.

- **Conjunto de servicios de Canal Seguro (SecureChannel):** Este conjunto incluye servicios para determinar la configuración de seguridad de un servidor y establecer un canal de comunicación en el que se garantiza la confidencialidad e integridad (integrity) de los mensajes intercambiados. Estos servicios no se aplican directamente en la aplicación OPC-UA, pero son proporcionados por la pila de comunicación utilizada.

- **Conjunto de servicio de Sesión:** Este conjunto de servicios define los servicios que se utilizan para establecer una conexión de capa de aplicación (una sesión) con un nombre de un usuario específico.
- **Conjunto de servicios de Gestión de Nodos (NodeManagement):** Estos servicios proporcionan una interfaz para la configuración de los servidores. Se permite a los clientes añadir, modificar y borrar nodos en el espacio de direcciones.
- **Conjunto de servicios de Vista:** El conjunto de vista del servicio permite a los clientes descubrir los Nodos navegando. La Navegación permite a los clientes buscar arriba y abajo de la jerarquía del servidor, o seguir las referencias entre los nodos. Esto permite al cliente explorar la estructura del espacio de direcciones.
- **Conjunto de servicios de Atributos:** El conjunto de servicios atributo se utiliza para leer y escribir valores de los atributos. Los atributos son características primitivas de nodos que se definen por OPC-UA.
- **Conjunto de servicios de Métodos:** Métodos representan las llamadas de función de los objetos. Ellos son invocados y regresan después de la finalización. El conjunto de servicios de métodos define los medios para invocarlos.
- **Conjunto de servicios para Monitorización de Items:** Este servicio se puede utilizar para determinar qué atributos del espacio de direcciones debe ser monitorizado para los cambios por parte de un cliente, o los eventos que el cliente está interesado.
- **Conjunto de servicios de Suscripción:** Puede ser utilizado para generar, modificar o eliminar mensajes para los Items Monitorizados.
- **Conjunto de servicios de consulta:** Estos servicios permiten al cliente a seleccionar nodos del espacio de direcciones en base a ciertos criterios de filtro.

2.6.3.3 Modelo de Información

Mientras que la norma OPC clásica tiene un meta-modelo muy elemental debido a que está formado por etiquetas en una jerarquía simple y sencilla, OPC-UA ofrece un meta-modelo de información rico con técnicas orientadas a objetos utilizando jerarquías

más elaboradas y con la posibilidad de crear instancias de herencia entre éstas, como se observa en la Figura 11.

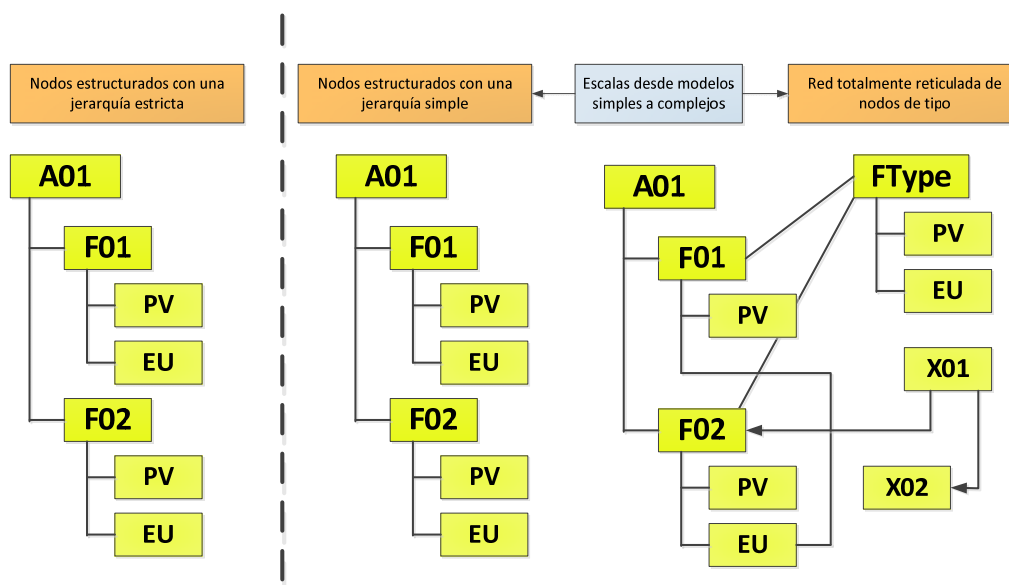


Figura 11: Comparación entre modelos OPC y OPC-UA

OPC-UA proporciona un meta-modelo estándar, conocido como Espacio de Direcciones, el cual se puede utilizar con el fin de modificarlo para definir Modelos de Información específicos. Está diseñado jerárquicamente y todos los niveles superiores de los servidores OPC-UA están diseñados de igual manera con esto se consigue la interoperabilidad entre ellos. El meta-modelo en OPC-UA define los objetos a utilizarse, tipos, variables, tipos de datos, etc [60]. Además, también posee información sobre el conjunto de instancias (u objetos) existentes en el sistema y el tipo de cada una de ellas. De este modo cualquier sistema, por complejo que sea, puede ser descrito en su totalidad mediante dichos mecanismos orientados a objetos. Por lo general, muy pocos sistemas empotrados implementarán todas las opciones de modelado descritas anteriormente.

Al usar meta-modelos, la información con una semántica conocida se puede intercambiar de una manera eficiente en lugar de datos simples. El modelo de información siempre está presente en un servidor OPC-UA, de esta manera el cliente no requiere tener integrado dicho modelo. La idea es que OPC-UA especifique cómo se intercambian los datos, mientras que los modelos de información estándar especifican qué información se

intercambia. El gran interés que suscita la modelización de la información ha generado el impulso para normalizar los modelos de información basados en OPC-UA. Esta norma define una serie de servicios web, los que permiten al cliente OPC-UA buscar y editar objetos en el espacio de direcciones de un servidor.

OPC-UA adicionalmente define un conjunto de servicios abstractos que puede ejecutarse en distintas infraestructuras de comunicación y utilizar el meta-modelo como base para definir los parámetros adecuados para los servicios. El modelo básico de información de OPC-UA proporciona los tipos de objetos primordiales y los puntos de acceso al espacio de direcciones del servidor. Sobre el modelo de información base se pueden establecer modelos de información normalizados o específicos del proveedor. [61]

OPC-UA ya define diversos modelos normalizados de información para el acceso de datos, alarmas y condiciones, programas, datos históricos y funciones agregadas. Como se puede ver en la Figura 12.

También proporciona los mecanismos para soportar múltiples modelos de información en un servidor. Esto permite proporcionar un alto nivel de interoperabilidad debido a que los datos no sólo pueden ser cambiados de una manera sencilla entre dispositivos, sino también con la semántica claramente definida.

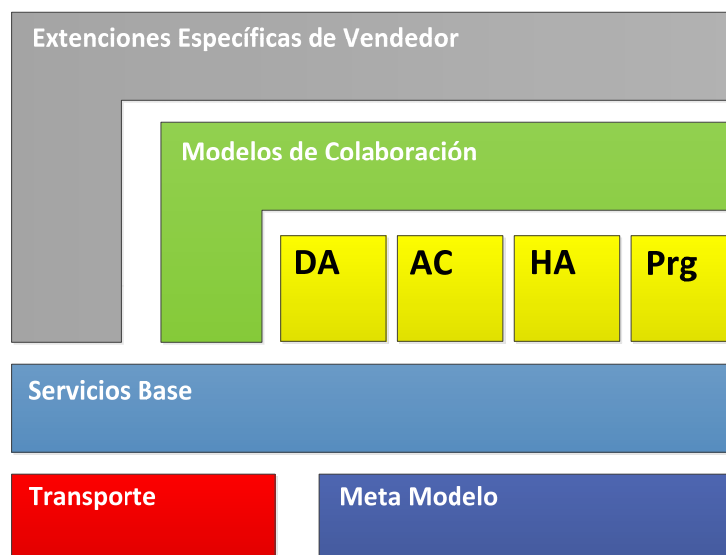


Figura 12: Capa de Modelos de OPC-UA

2.6.3.4 Modelo de Datos

El espacio de direcciones se compone principalmente de un conjunto de nodos los cuales están conectados por ciertas referencias. Los nodos se agrupan en clases basadas en sus significados. Cada clase de nodos incluye atributos que los describen en detalle. El modelo de nodos de OPC-UA permite que toda la información pueda ser conectada de diferente manera por medio de referencias de tipo jerárquicas o no jerárquicas. Esto facilita la exposición de la información en diferentes formas dependiendo de su uso. Las clases principales se representan en la Figura 13 y se describen a continuación: [62]

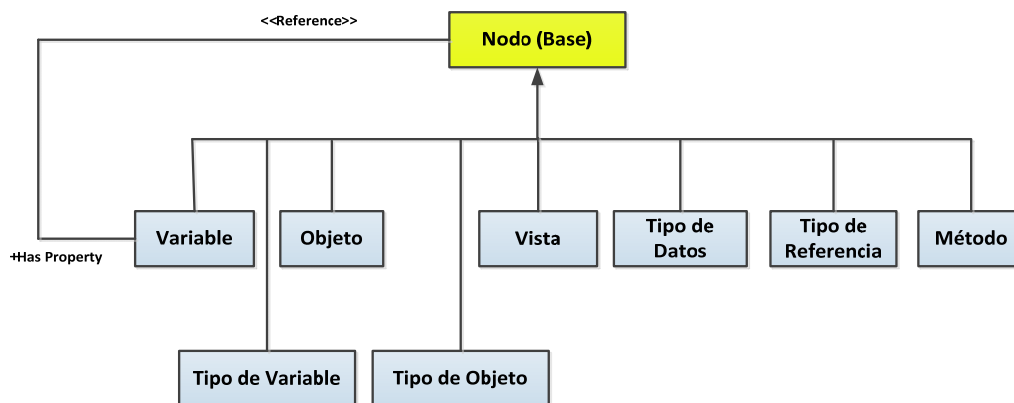


Figura 13: Resumen de los principales nodos en OPC-UA

- **Nodo:** Son los componentes fundamentales del espacio de direcciones del servidor. Describen los atributos y referencias que deben ser instanciadas cuando un Nodo es definido. Los atributos son los únicos elementos que contienen los valores de los datos. Los atributos son componentes elementales de los nodos, sus valores en un servidor OPC-UA se pueden tener acceso por un cliente usando servicios de lectura, escritura, consulta o suscripción/monitorización.
- **Clases Nodos (NodeClass):** Definen los metadatos para los componentes de un modelo de objeto de OPC-UA. Ellos también definen un conjunto de atributos fijos que no pueden ser ampliados ya sea por el servidor o un cliente.
- **Tipos de Objetos (ObjetcType):** Son Nodos que representan un tipo de definición para un Objeto.

- **Objeto (Object):** Son Nodos que representan un elemento físico o abstracto de un sistema. Los Objetos son modelados usando el modelo de objetos de OPC-UA. Los sistemas, sub-sistemas y dispositivos son ejemplos de Objetos.
- **Referencia (Reference):** Es una relación explícita de un Nodo con otro. El Nodo que contiene la referencia es llamado el Nodo Fuente (sourceNode) y el Nodo referenciado es el Nodo Objetivo (targetNode). Las referencias se pueden acceder indirectamente usando servicios de búsqueda o consulta.
- **Tipo de Referencia (ReferenceType):** Son Nodos que representan un tipo de definición para una Referencia. Define la semántica de las relaciones entre los Nodos Fuentes y Nodos Objetivos.
- **Tipos de variables (variableTypes):** Son Nodos que representan el tipo de definición para una Variable.
- **Variables:** Es un Nodo que contiene un valor. Existen dos tipos:
 - Variables que describen las características de un Nodo, a estas se las denominan **Propiedades**.
 - Variables que representan el contenido de un objeto, son denominados **Datos**.
- **Tipo de Datos (DataType):** Se usa en conjunto con los atributos de las variables y las definen completamente.
- **Vistas (Views):** Define cierta parte del Address Space de esta manera se proporciona solo los nodos usados por el usuario final.
- **Métodos (Method):** Define la característica de un método, el cual puede ser ejecutado a través de las interfaces OPC-UA.

La utilización de las técnicas orientadas a objetos incluye las jerarquías y las herencias, lo que permite que la información sea expuesta de manera transparente y se puede tener acceso de diferentes maneras. Los nodos forman una red totalmente conectada que permite que la información pueda ser compartida por todos los usuarios. No existen límites en cómo se diseña el modelo de información ya que posee varios mecanismos para su extensión. El modelado de información siempre ocurre a nivel del servidor, y podrá, a

largo plazo, reducir drásticamente los costes técnicos integrando sistemas que utilizan productos de distintos proveedores. Las especificaciones de OPC-UA dan los conceptos básicos para el modelado de la información y el objetivo adicional es que los vendedores propietarios de software o las organizaciones de normalización desarrollen sus modelos de información.

2.6.3.5 Seguridad

La seguridad es un requisito fundamental para OPC-UA y por lo tanto se encuentra integrada dentro de la arquitectura. Los mecanismos se basan en un análisis detallado de las amenazas. OPC-UA maneja la seguridad con la autenticación de los clientes y los servidores, la integridad y la confidencialidad de los mensajes intercambiados y la verificabilidad de los perfiles funcionales de equipos.

Se basa en la arquitectura mostrada en la Figura 14. Los tres niveles de seguridad son: seguridad del usuario, seguridad de las aplicaciones y la seguridad en el transporte de datos.

Los mecanismos de seguridad a nivel de usuario de OPC-UA se ejecutan una vez cuando una sesión está configurada. El cliente transmite una señal de seguridad encriptada, que identifica el usuario, al servidor. El servidor autentica al usuario basándose en el token y luego autoriza el acceso a los objetos en el servidor. La norma OPC-UA no define mecanismos de autorización tales como listas de control de acceso.

OPC-UA posee seguridad a nivel de aplicación y es también parte de la configuración de sesión e incluye el intercambio de certificados firmados digitalmente. Existen dos tipos que son: los *certificados de instancia* que identifica la instalación concreta del software de comunicación y los *certificados de software* que identifican el cliente y el servidor de software y los perfiles de OPC-UA implementados. Describen capacidades del servidor, tales como el apoyo a un modelo de información específico. [63]

Los certificados de seguridad de nivel de transporte se pueden utilizar para proporcionar integridad mediante la firma de los mensajes y la confidencialidad mediante el

cifrado de los mismos. Esto evita que la divulgación de la información intercambiada y asegura que los mensajes no han sido manipulados.

Los mecanismos de seguridad se realizan como parte de las pilas de OPC-UA, es decir, que se incluyen en un paquete de software proporcionado por la Fundación OPC y están listas para ser utilizado por el cliente y el servidor.

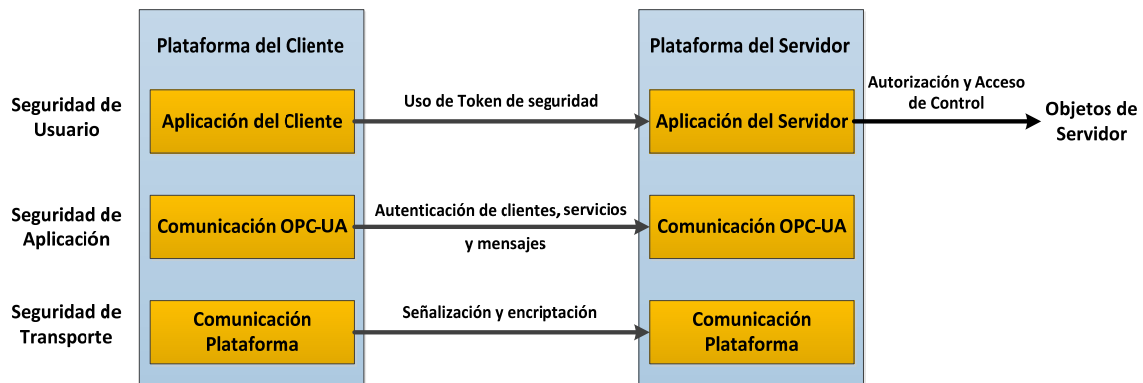


Figura 14: Concepto de Seguridad Escalable

En resumen, OPC-UA ofrece una solución completa que permite satisfacer los requisitos de todas las capas verticales para el acceso a dispositivos remotos. Brinda una infraestructura eficiente y segura para las comunicaciones entre sensores y controladores en los sistemas de automatización y control de fabricación y procesos en las empresas. Provee comunicaciones seguras utilizando estándares de la industria informática. Esto permite que multitud de dispositivos puedan recibir y convertir los datos remotos en información útil para tomar decisiones inteligentes. Proporcionan una infraestructura para múltiples proveedores, la interoperabilidad multiplataforma y seguridad para los entornos de automatización industrial.

2.7 ESTÁNDARES DE AUTOMATIZACIÓN DE SISTEMAS DISTRIBUIDOS

A pesar de que los conceptos y la sintaxis es la misma para todas las herramientas de programación IEC 61131, sin embargo, la semántica de los elementos del lenguaje está definido de manera ambigua en el apartado IEC 61131-3. Es por esto, que las herra-

mientas de software interpretan el estándar de manera distinta lo que resulta en una ejecución completamente diferente usando el mismo código. Por lo tanto, no es posible transferir la configuración de una herramienta a otra y de esta manera preservar toda la información requerida para una ejecución correcta del algoritmo de control.

En los procesos de fabricación actual se busca conseguir un alto grado de distribución del control. La norma IEC 61131 no es adecuada para diseñar, implementar y ejecutar sistemas de control en nuevos componentes hardware como: sistemas empotrados, PC industriales y redes más avanzadas y a la vez no pueden ser utilizados para descentralizar la lógica de control y poder implementar controles holónicos o multi-agente, que hagan los sistemas más flexibles. Además, la creciente importancia del software en el control hace que la reutilización de código sea vista como una cuestión fundamental. Es decir, poder diseñar un mismo software de control para dos máquinas que ejecutan el mismo proceso, pero son de fabricantes diferentes.

Por los motivos mencionados anteriormente, es necesaria la implementación de un nuevo estándar el cual ofrezca una vista complementaria y una solución eficaz a problemas similares o más complejos de los que la norma IEC 61131 podría resolver y es por esta razón que se ha diseñado la norma IEC 61499.

2.7.1 Estándar IEC 61499

Fue creado para sistemas de control distribuido, incluyendo su arquitectura y los requisitos de herramientas de software. Se desarrolló como consecuencia del creciente interés en las nuevas tecnologías y arquitecturas para crear la próxima generación de sistemas industriales y teniendo como base el estándar IEC 61131. IEC 61499 fue diseñado por el comité técnico TC-65 de medida, control y automatización de procesos industriales (TC, Technical Committee), que pertenece a la IEC, siendo aprobada la primera versión en Agosto de 2005 [64]. Esta norma define una arquitectura genérica y una guía para el uso de los Bloques Funcionales (FBs) en Sistemas de Control y Medición de Procesos Industriales Distribuidos (IPMCSs).

Uno de los principales objetivos de IEC 61499 es promover el desarrollo de sistemas heterogéneos compuestos de dispositivos de control de diferentes fabricantes permitiendo la reconfiguración dinámica, es decir, cambiar la configuración de un sistema mientras la aplicación de control continúa ejecutándose.

IEC 61499 es visto como la siguiente generación de estándares en sistemas de automatización y está diseñado para cubrir interoperabilidad, portabilidad y reconfigurabilidad, que no están contemplados en IEC 61131-3. Por el momento, en la práctica industrial son pocos los sistemas basados en IEC 61499, pero actualmente, una gran cantidad de trabajos de investigación aceptan y utilizan los conceptos básicos del estándar.

2.7.1.1 Especificaciones IEC 61499

El estándar IEC 61499 se divide en los siguientes 4 apartados: [65]

- a) **Arquitectura.** IEC 61499-1 contiene los requisitos generales, las definiciones y modelos de referencia. Adicional a esto posee las reglas para la declaración de los tipos de FBs y reglas para su comportamiento.
- b) **Requisitos de herramienta software.** IEC 61499-2 define los requisitos de las herramientas software utilizadas para esta norma, que soporten la ejecución de las tareas de ingeniería en los sistemas de control distribuido y la especificación de los tipos de FBs.
- c) **Manual Informativo.** IEC 61499-3 contiene la información para el entendimiento, aceptación y aplicabilidad, tanto de la arquitectura IPMCS, como de herramientas software que cumplan con las especificaciones del estándar.
- d) **Reglas y Perfiles de Conformidad.** IEC 61499-4 contiene la definición de las reglas para el desarrollo de perfiles de conformidad con la norma, las cuales especifican las características para implementar los apartados 1 y 2.

2.7.1.2 Arquitectura

IEC 61499 define una arquitectura genérica y jerárquica de modelos, permitiendo entender la organización del sistema y sus componentes. Desarrolla una nueva estructura

para aplicaciones de control distribuido. Los modelos son genéricos, independientes del dominio y extensibles con la definición y uso de FBs. Los modelos son: Modelo de Bloque Funcional (FB), Modelo de Recurso, Modelo de Dispositivo, Modelo de Sistema, Modelo de Aplicación, Modelo de Distribución y Modelo de Gestión.

A) Modelo de Bloque Funcional (FB)

El FB es el elemento más pequeño en un sistema de control distribuido. El FB está formado por dos partes: cabeza y cuerpo. La primera consiste en una cabeza que está conectada al flujo de eventos. Acepta eventos de entrada y genera eventos de salida, como se representa en la Figura 15. La segunda parte es el cuerpo, el cual está conectado al flujo de datos, acepta los datos de entrada y genera datos de salida. El comportamiento dinámico del FB está definido por la Gráfica de Control de Ejecución (siglas de: ECC, Execution Control Chart) que procesa las entradas de eventos y genera salidas de eventos. [66]

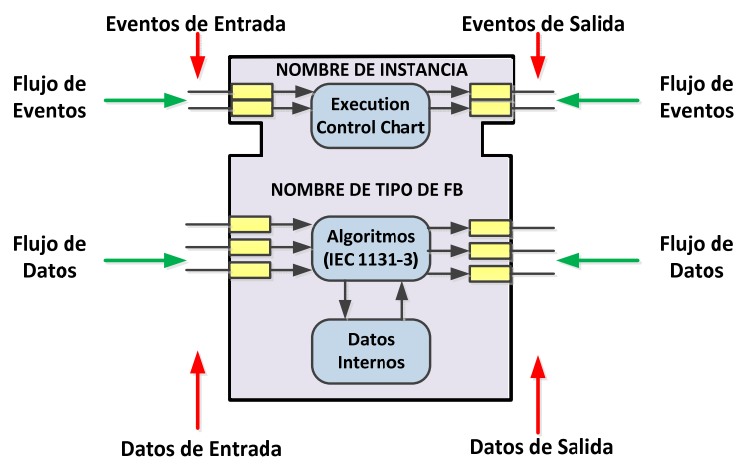


Figura 15: Modelo de Bloque de Función

Un FB en la norma IEC 61499 se mantiene pasivo hasta que es disparado por un evento de entrada, es decir, todos los eventos de entrada son usados para activar un bloque funcional. El FB ejecuta y produce eventos y datos de salida como se representa en la Figura 15. El ECC describe el comportamiento interno de las instancias de los FBs básicamente.

cos. Ayuda al programador a descomponer el comportamiento complejo en pequeñas partes llamados estados. Cada estado es válido bajo un cierto conjunto de condiciones. Los estados son asociados con uno o más algoritmos y/o con eventos de salida. La activación del estado implica la ejecución de los algoritmos adjuntos.

La funcionalidad del FB esta proporcionada por medio de algoritmos. Un algoritmo puede estar escrito en cualquiera de los 5 lenguajes que menciona el IEC 61131-3: IL, ST, LD, FBD y SFC. También en otros lenguajes de alto nivel como: C, C++, Java y Delphi. El algoritmo procesa entradas y datos internos y/o externos, generando datos de salida. Las variables internas o la información de estado no son accesibles por el flujo de datos.

IEC 61499 define tres diferentes tipos de Bloques de Función: [67]

- **FB Básico:** Es la unidad más pequeña de programación. Consta de dos partes: ECC y algoritmos.
- **FB Compuesto:** Compuesto por una red de instancias de FBs interconectados entre sí.
- **FB Interfaz de Servicio:** Proporciona servicios a una aplicación, como interacción entre aplicación y recursos, comunicación, etc.

B) Modelo de Recurso

Un recurso es una unidad funcional con control independiente de operación, que proporciona servicio a las aplicaciones, incluyendo la planificación y ejecución de los algoritmos [68]. Las funciones son:

- Aceptar los eventos y/o los datos de entrada/salida de las interfaces de proceso y comunicaciones.
- Procesar los eventos y/o los datos, enviar los eventos y/o los datos a las interfaces de proceso y comunicaciones, como se indica en la Figura 16.

El recurso en esta norma está modelado por tres elementos:

- **Aplicación Local** (o parte local de aplicación distribuida): Posee variables y

eventos de entrada y salida de los diferentes Bloques de Función que ejecutan las operaciones necesarias dadas por la aplicación.

- **Interfaz de Proceso:** Su principal objetivo es ejecutar un mapeo de eventos y datos entre las aplicaciones e interfaces de proceso, esto se logra mediante el uso del Bloque de Función de Interfaz de Servicio (SIFB).
- **Interfaz de Comunicación:** Al igual que la interfaz anterior, su función es realizar el mapeo de eventos y datos entre las aplicaciones e interfaces de comunicaciones. Se lleva a cabo con los SIFBs.

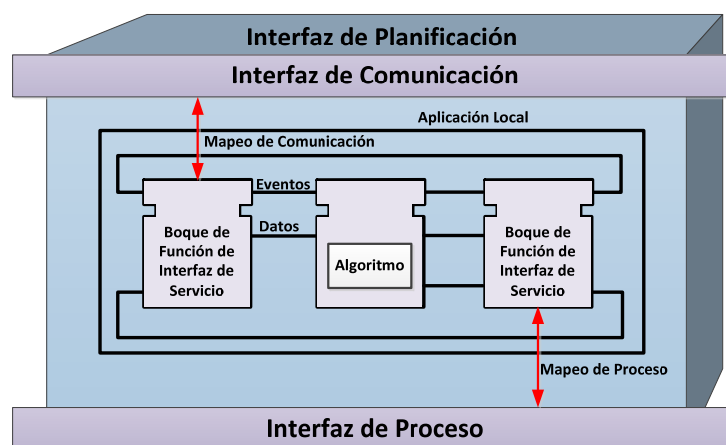


Figura 16: Modelo de recurso

C) Modelo de Dispositivo

Un dispositivo es una entidad física independiente, capaz de realizar una o más funciones específicas en un contexto particular delimitado por sus interfaces (de proceso y de comunicación)[69]. Se considera un contenedor de recursos que proporciona un entorno de ejecución para aplicaciones. Un dispositivo puede ser conectado a más de un segmento. Podemos notar que posee dos tipos de interfaces:

- **Interfaz de Proceso:** permite la comunicación entre otros dispositivos y aplicaciones.
- **Interfaz de Comunicación:** Logra la comunicación entre los dispositivos y aplicaciones del proceso.

Su modelo se representa en la Figura 17.

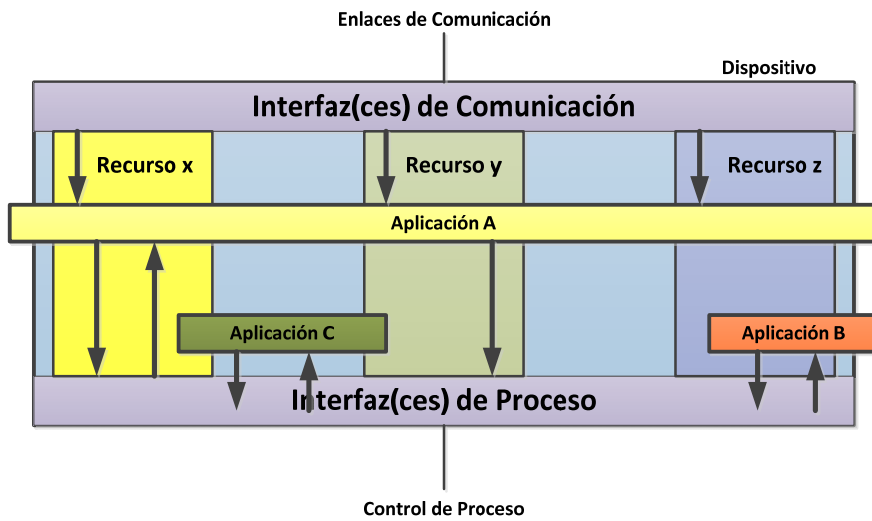


Figura 17: Modelo de dispositivo

D) Modelo de Sistema

Un sistema consiste en una colección de dispositivos interconectados y comunicados entre sí, por medio de una red de comunicaciones a través de segmentos y enlaces para formar un conjunto de cooperación de aplicaciones. Describe un segmento de red de un cierto tipo, al cual varios dispositivos son conectados a través de enlaces. Se puede modelar como se representa en la Figura 18.

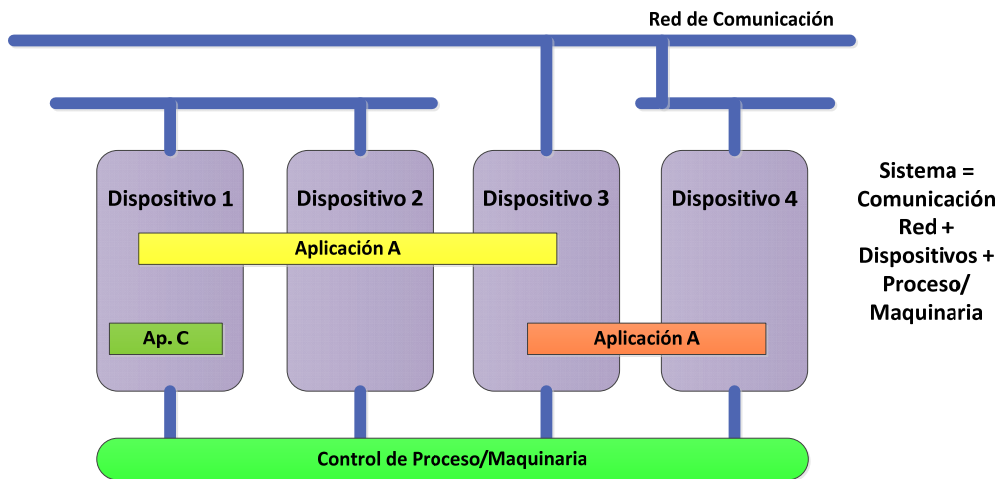


Figura 18: Modelo de sistema

E) Modelo de Aplicación

Una aplicación es una unidad funcional de software específica para la solución de un problema en medición de procesos industriales o de control. Puede distribuirse entre varios recursos, en el mismo o en diferentes dispositivos (subaplicación) y puede comunicarse con otras aplicaciones. Usa las relaciones especificadas por la aplicación para determinar la respuesta apropiada de eventos entre las interfaces de proceso y comunicación. Usando la programación y ejecución de algoritmos internos permite la modificación de variables, generación de eventos adicionales e interacciones con interfaces de proceso y comunicación. Cada aplicación está formada por una red de FBs, especificando el flujo de datos y eventos entre las FBs, como se representa en la Figura 19.

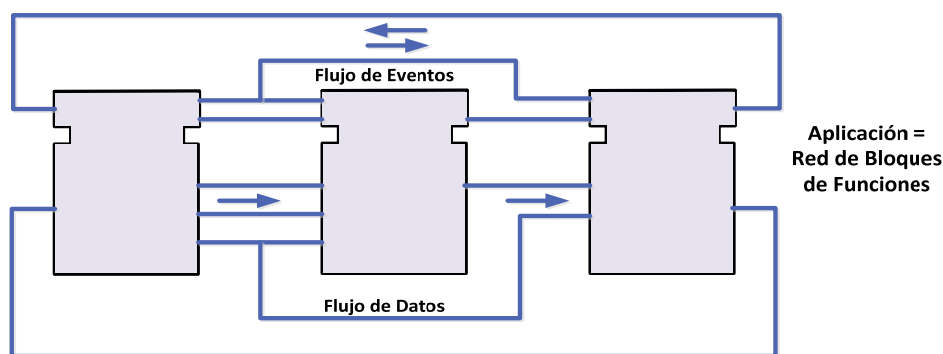


Figura 19: Modelo de aplicación

F) Modelo de Distribución

La fase final del proceso de desarrollo de la aplicación en IEC 61499-1 es la distribución de la aplicación de control entre los dispositivos de control. En este paso los FBs de la aplicación serán mapeados a los dispositivos de control donde serán ejecutados. Una aplicación puede ser distribuida colocando las instancias de los FBs que forman la aplicación sobre los diferentes recursos en uno o más dispositivos [70]. Este modelo se representa en la Figura 20.

G) Modelo de Gestión

Proporciona herramientas para la gestión de la relación de los recursos con los dispositivos. El estándar propone dos esquemas: [71]

- Primer esquema: presenta la gestión de recursos compartidos que proporciona facilidades para la gestión de otros recursos dentro de un dispositivo.

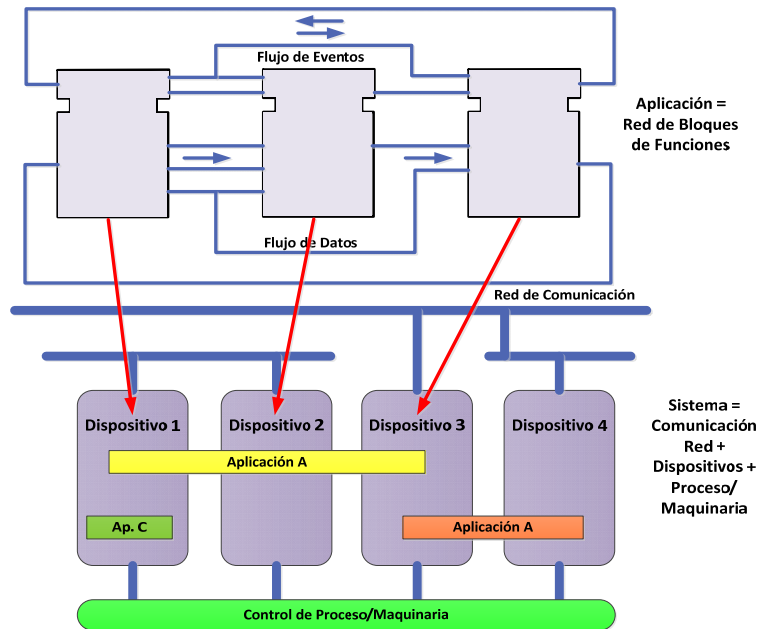


Figura 20: Modelo de distribución

- Segundo esquema: presenta la gestión de servicios de distribución de recursos dentro de un dispositivo.

La configuración de un IPMCS distribuido basado en IEC 61499 puede ser permitida por el uso de funciones de gestión, las cuales pueden ser incluidas en cada dispositivo. Para este propósito el estándar define un dispositivo de gestión y su interfaz, que es un tipo de FB de gestión.

2.7.2 Entornos de Desarrollo y de Ejecución de la Norma IEC 61499

El entorno de desarrollo integrado, llamado también IDE es la herramienta de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI) para el diseño de sistemas de automatización bajo norma IEC 61499.

Un entorno de ejecución (RTE, Runtime Environment) es una máquina virtual el cual proporciona un conjunto de bibliotecas, servicios y otros componentes necesarios para que una aplicación IEC 61499 pueda ser ejecutada. El RTE actúa como un intermediario entre el sistema operativo y aplicaciones IEC 61499.

En la siguiente Tabla 3 se muestran los principales sistemas de desarrollo (herramienta software) y sus entornos de ejecución bajo licencia de código abierto (open source) que se utilizan actualmente para el diseño de aplicaciones IEC 61499:

Tabla 3: *Sistemas de Desarrollo y Ejecución de IEC 61499*

Entorno de Desarrollo (IDE)	Entorno de Ejecución (RTE)
4DIAC-IDE	FORTE
FBDK	FBRT
ISaGRAF Workbench	ISaGRAF Runtime
nxtSTUDIO	nxtRT61499F ³²

A continuación, se describe los entornos de desarrollo y programación que son productos comerciales compatibles o disponible como código abierto para IEC 61499 en especial 4DIAC y FBDK ya que son los más utilizados a nivel académico.

2.7.2.1 FBDK / FBRT

FBDK, kit de desarrollo de bloques funcionales (FBDK, Function Block Development Kit), es una herramienta de soporte de ingeniería, para desarrollo de software basada en IEC 61499.

FBDK es de libre distribución, se ajusta al propósito de educación y es usado para proyectos de investigación [72]. FBDK fue originalmente desarrollado por el Advanced Technology Division of Rockwell Automation en el año 2005. Contiene un editor de bloques funcionales (FBEditor, Function Block Editor) y un entorno de ejecución (FBRT, Function Block RunTime).

FBDK permite ingeniería centrada en la aplicación, tiene una librería de componentes de software extensa y se puede descargar la aplicación de control a diferentes dispositivos hardware.

A) Entorno de desarrollo: FBDK

El editor de FBDK es un IDE que soporta desarrollo gráfico de FBs, sistemas y traducción de clases java. Almacena los FBs en el formato basado en lenguaje de marcado extensible (XML, eXtensible Markup Language) definido en el apartado 2 del estándar IEC 61499. Se pueden desarrollar y desplegar aplicaciones de sistemas de control distribuido basadas en FB.

FBDK permite a los usuarios definir, construir y probar tipos de FBs, tipos de recurso, tipos de dispositivo y configuración de sistemas. Puede usar los FBs incorporados o usar FBs definidos por el usuario para diseñar los elementos de una aplicación de control distribuida.

Usando FBDK, cada tipo de FB puede ser traducido a una clase en java. Cada uno de los FBs hecho por el usuario o incorporado, puede ser probado por la seguridad de sus funcionalidades diseñadas. Sin embargo, en aplicaciones centralizadas como distribuidas pueden ser ejecutados con una interfaz gráfica para visualización y variación de parámetros.

Cada modelo de dispositivo contiene un recurso con nombre Manager (MGR) el cual es de tipo RMT_RES. Estos recursos en coherencia con FBRT utilizan la aplicación de control distribuida para ejecutarse en diferentes controladores.

La comunicación entre el FBEditor y el FBRT es implementada usando una comunicación abierta, protocolo de administración basado en XML. FBDK tiene una copia integrada de FBRT, de forma que las aplicaciones pueden ser ejecutadas directamente desde el entorno integrado sobre el mismo dispositivo donde FBDK se ejecuta. FBDK también ejecuta configuraciones de sistemas en varios dispositivos distribuidos.

Las características más relevantes de FBDK son:

- Es una herramienta software para desarrollar sistemas de control a partir de FBs.
- Se programa gráficamente y textualmente.
- Tiene librería de FBs del estándar IEC 61499.

- Permite crear y modificar la interfaz (FBs).
- Permite modificar la gráfica de control de ejecución (ECC).
- Se compila en Java.
- Se ejecuta bajo Windows y Linux.
- El FBEditor almacena los FBs en formato XML.

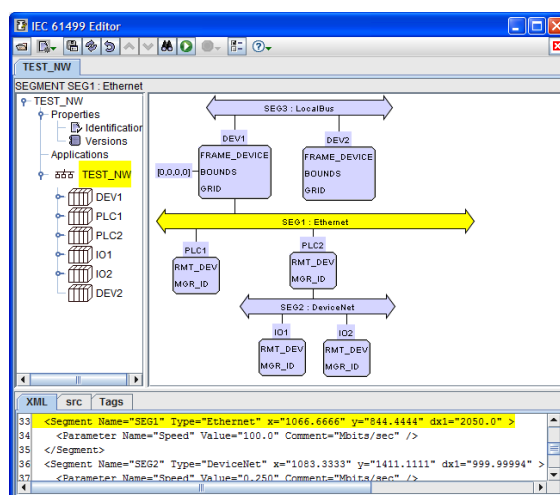


Figura 21: Sistema de configuración en FBDK

B) Entorno de ejecución: FBRT

El entorno de ejecución de FBDK es el FBRT. Es un software que se ejecuta usando la máquina virtual de Java en una computadora personal o en otra plataforma hardware que soporte Java. Utiliza entorno de ejecución Java (JRE, Java Runtime Environment) para ejecutar dispositivos remotos.

El tiempo real de ejecución basado en la máquina virtual de JAVA, no se puede adaptar a todas las aplicaciones requiriendo de respuesta rápida del controlador. Usa el modelo recurso multi-tareas no preferente (NPMTR, Nonpreemptive Multi-Threaded Resource) para su ejecución, el cual tiene implementado.

En el FBRT los eventos de salida que disparan la invocación de otros bloques, son implementados como una serie de llamadas a funciones directas para el bloque destino dentro de una simple tarea. Este mecanismo detiene la ejecución en la llamada al bloque

a otros FBs a lo largo de la ruta de propagación de eventos hasta que su ejecución haya sido completada.

Es importante resaltar dos características referentes al FBDK:

1. El FBDK no soporta el paradigma de ingeniería centrado en la aplicación.
2. El usuario tiene que agregar necesariamente los SIFBs de comunicaciones manualmente.

2.7.2.2 4DIAC-IDE / FORTE

4DIAC Estructura para la Automatización y Control Industrial Distribuida (siglas en inglés de Framework for Distributed Industrial Automation and Control), es una herramienta de ingeniería de código abierto basada en la plataforma de Eclipse, para automatización distribuida, reconfigurable y software de control. 4DIAC fue creada en el año 2000 por PROFACTOR GmbH & Viena University of Technology. [73]

A) Entorno de desarrollo: 4DIAC-IDE

El editor es un entorno de desarrollo integrado denominado 4DIAC-IDE. El objetivo de la iniciativa de 4DIAC es proporcionar herramientas conforme al estándar que permiten establecer un entorno de automatización y control, basado en los objetivos de portabilidad, configurabilidad e interoperabilidad, mismos que son mencionados en el IEC 61499. 4DIAC persigue los siguientes objetivos: [67]

- Suministrar una base común para desarrollo, dispositivo industrial e investigación de IEC 61499.
- Suministrar un paquete conteniendo un entorno runtime para diferentes plataformas de control empotradas y el entorno de ingeniería.
- Suministrar ejemplos reales a nivel prototipo para incrementar la aceptación del IEC 61499 en la industria.
- Proporcionar un incentivo para el uso de IEC 61499 con la industria.

Las características más relevantes de 4DIAC son:

- 4DIAC-IDE es una herramienta IEC 61499 basada en Eclipse.
- Tiene tipos de datos elementales de acuerdo a IEC 61131-3.
- Tiene conexiones de eventos y datos.
- Configuración de comandos (crear, escribir, iniciar de acuerdo con IEC 61499).
- Tiene FBs de comunicación (Client/Server, Publish/Subscribe para Ethernet).
- Puede ejecutar bloques funcionales de tipo FB básicos, FB compuestos, FB interfaces de servicio SIFBs y adaptadores.
- Se ejecuta bajo plataforma Windows, Linux y Solaris.

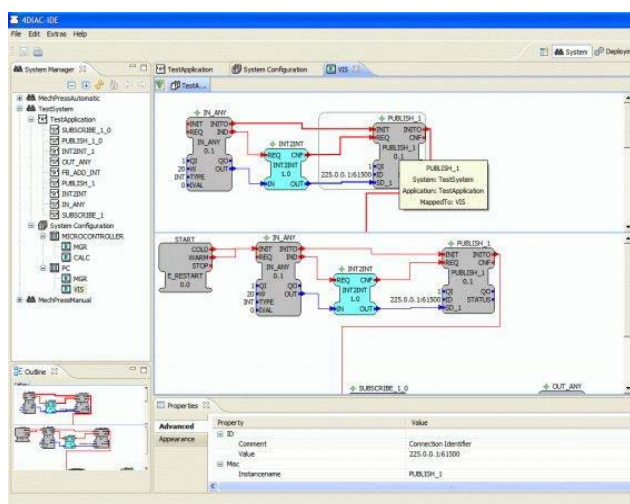


Figura 22: Sistema de configuración en 4DIAC-IDE

B) Entorno de Ejecución: FORTE

El entorno de ejecución de 4DIAC conforme a IEC 61499 es 4DIAC-RTE (FORTE). FORTE es una pequeña implementación portable de un entorno runtime conforme a IEC 61499 enfocado a pequeños dispositivos de control empotrados (16/32 Bit), implementado en C++ y es portable para múltiples plataformas[74]. Los mecanismos de ejecución en FORTE permiten la ejecución limitada en tiempo real de configuraciones de control IEC 61499 desencadenadas por eventos externos, donde las diferentes partes de la configuración pueden cumplir diferentes limitaciones en tiempo real y la ejecución de

los procesos de baja prioridad no perturban la ejecución de los procesos de mayor prioridad.

En el entorno runtime FORTE hace uso de un disparador de eventos para la planificación de FBs. El planificador presenta un enfoque de colas donde todos los eventos de entrada se los entrega a los FBs destino en orden FIFO (primero entra-primero sale). El disparador de eventos desacopla la ejecución de envío de eventos FB del bloque receptor, de ese modo crea el periodo de bloqueo de un FB independiente de la topología de red. [75]

El entorno de ejecución sigue en continuo cambio en lo referente a la optimización de ejecución e implementación de interfaz de comunicaciones. FORTE ha sido escrito para funcionar independientemente de la plataforma a utilizarse, esto permite hacer más fácil su uso en diversos tipos de hardware y plataformas de sistemas operativos. La versión actual de FORTE es soportada por los siguientes sistemas: Windows(Win32), eCos, POSIX, Lego Mindstorms NXT controller, etc.

2.8 TRABAJOS RELACIONADOS

En esta sección se presentan aquellos trabajos que han introducido algún método y/o tecnología para implementación de CPPS bajo IEC 61499 en el sector industrial. El objetivo principal en todos los casos estudiados es mejorar la modularidad y reutilización, facilitando por tanto el desarrollo y mantenimiento de las aplicaciones complejas cada vez más demandadas por este sector.

En los siguientes apartados se estudian trabajos de otros autores en los que se utilizan comunicaciones basadas en OPC-UA para aplicaciones de control industrial. También se evalúan entornos de soporte al proceso de desarrollo de aplicaciones, no necesariamente del sector de la industria manufacturera.

2.8.1 Diseño de Sistemas CPPS bajo norma IEC 61499

En esta sección se analizan las investigaciones relacionadas directamente con las áreas de mecanizado y producción industrial en las cuales se ha utilizado la norma IEC

61499. También se describe el enfoque y alcance de la investigación propuesta en este artículo en el cual se desarrolla una metodología basado en IEC 61499 y en una Arquitectura CPPS para la planificación de procesos distribuidos y adaptable a procesos continuos, especialmente en el área de petróleo y gas.

Investigaciones sobre el diseño, uso e implementación de Bloques de Función (FBs) usando la norma IEC 61499 en diferentes aplicaciones de control se han realizado en los últimos años por las universidades y centros de investigación a nivel mundial. La mayoría de estas investigaciones se han limitado a realizar control básico con supervisión de PLCs en un proceso industrial, pero estos dispositivos tienen una desventaja, que no son capaces de manejar problemas de incertidumbre, así como, el diseño y proceso de planificación a nivel de fábrica en un proceso de producción industrial. De acuerdo con la bibliografía consultada, los FBs diseñados en IEC 61499 tienen un uso limitado para procesos adaptativos y control de sistemas básicos de mecanizado/montaje; por este motivo, el objetivo de esta investigación es utilizar la arquitectura de FBs en un proceso continuo real.

Se han diseñado e implementado aplicaciones industriales usando la arquitectura de FBs dada por la norma IEC 61499 para el control de sistemas distribuidos integrando algoritmos inteligente como: lógica difusa o redes neuronales en dispositivos empotrados [76], otras aplicaciones industriales que se han beneficiado de la norma IEC 61499 son: el mantenimiento predictivo de sistemas de control distribuido [77], la verificación automatizada de sistemas de control industrial basados en FBs [78], la ingeniería de sistemas de soporte orientados a FBs [79] y los modelos de FB concurrentes reconfigurables. [80]

Estudios sobre la implementación de IEC 61499 en sistemas de control de procesos se presentan en Olsen [81]. En este artículo se diseña e integra modelos en tiempo real usando una plataforma basada en Java, donde una aplicación de control se distribuye a través de dos dispositivos usando el software MANAGER FB, la cual es capaz de proporcionar servicios de gestión para dispositivos de control.

Zoitl [82] muestra la ejecución de aplicaciones en tiempo real basadas en IEC 61499, describiendo los elementos de ejecución dentro de un dispositivo y diferentes enfoques de programación e implementación. Strasser [83] realiza críticas e indica claramente las ambigüedades en la implementación de la norma IEC 61499, lo que conduce, a diferentes comportamientos en la ejecución de las FBs y finalmente da soluciones a estos problemas.

El diseño, implementación y uso de librerías de FBs para control empotrado bajo la norma IEC 61499 se indica en Strasser [84] y demuestra su uso en un experimento real y conocido como es el control de un balancín. [85], [86], [87] ha desarrollado una aplicación llamada EMBENCH. Se trata de un entorno de diseño para simulación y prototipado rápido en diferentes niveles para el control de mecanizado, utilizando IEC 61499 para modularización y reutilización de los servicios implementados para el control. El sistema permite programar diferentes tipos de comandos en los controladores que se utilizan en diferentes etapas de producción es decir permiten programar funciones básicas desde servomotores hasta complejas celdas de fabricación flexible. Hussain et al [88], demuestran un ejemplo de cómo se puede utilizar IEC 61499 para modelar una aplicación distribuida, flexible y reconfigurable. Pero hasta el momento, la norma IEC 61499 sólo se ha centrado en los procesos de control discreto, esta investigación utiliza IEC 61499 para integrar la comunicación industrial y su uso en la monitorización de procesos analógicos.

El uso de la arquitectura FB para la planificación de procesos ha surgido como un enfoque innovador en los sistemas de fabricación, principalmente desde la introducción de la norma IEC 61499. Con el nuevo estándar, los FB pueden ser activados por eventos para ejecutar algoritmos internos de una manera controlada. Esta característica abre nuevos escenarios de aplicaciones interesantes. La posibilidad de manejar los cambios durante la generación y ejecución de los planes de proceso, así como la vinculación con la programación para un rendimiento óptimo del sistema, son efectos desafiantes de la naturaleza distribuable y modular de la tecnología FB.

Dentro de esta perspectiva, los FBs aplicados a la planificación del proceso de ensamblaje se describen en Wang et al [89]. En dicho artículo se identifican las características del montaje de componentes (pares de componentes de acoplamiento) y se asignan a los FBs apropiados para su correcto uso. Cada FB posee un conjunto de algoritmos para determinar cómo realizar una operación de montaje la cual puede ser realizada por un robot o un trabajador humano. Con el fin de facilitar el diseño de FBs [90] desarrolló una aplicación para diseñar FBs, la cual permite generación de planes de procesos adaptativos y capaz de encapsular planes de procesos genéricos en FBs para ejecución en runtimes específicos.

Otro artículo interesante es el propuesto por Vicaire et al [91], en el cual se proponen metodologías de programación para CPPSs basadas en modelos. El modelo de programación dado en este artículo es llamado “Bundle” y es adecuado para modelar tanto dispositivos de nivel de planta como sensores y actuadores. Sin embargo, no se atiende su integración con normas de automatización como IEC 61499 o IEC 61131.

En resumen, estas investigaciones se basan en la coincidencia de las técnicas de búsqueda de nuevas metodologías para producir sistemas distribuidos bajo la norma IEC 61499, pero no se centran en la integración con un protocolo de comunicación útil en fábricas inteligentes, tal como OPC-UA podría ser.

2.8.2 Comunicaciones M2M para sistemas CPPS

En este punto se presenta un conjunto de trabajos relacionados, paradigmas y tecnologías de desarrollo, que proporcionan arquitecturas orientadas a servicios y sistemas M2M, empleando el estándar IEC 61499.

Stojmenovic et al [92] considera a M2M como una tecnología clave para CPPS. El autor identifica el problema de que todas las investigaciones existentes en el ámbito de la comunicación M2M se basan en modelos a pequeña escala y soluciones centralizadas. En este sentido, propone un cambio de paradigma y sugiere que los nodos también deben tomar decisiones basadas en el conocimiento local, en lugar de sólo el reenvío de mensajes al sistema central. Usando este enfoque en sistemas de producción permitiría una fácil

vinculación de los datos de planta, a partir de un conjunto de variables que se transmiten de una manera segura y en donde el control industrial se realiza también localmente. Sin embargo, en este artículo no se menciona cuáles serían las ventajas de aplicar esta arquitectura con normas creadas para la su aplicación en sistemas distribuidos como IEC 61499, o desde el punto de vista de las comunicaciones, la utilización de normas como OPC-UA que podrían brindar comunicaciones seguras.

Kang et al. [93] proponen un middleware llamado Servicio de Distribución de Datos en Tiempo Real (RDDS) para CPPSs; este servicio facilita el intercambio de variables de una manera segura en este tipo de sistemas. El mecanismo de publicación/suscripción que se diseña e integra en el middleware tiene como fin el mejorar la fiabilidad y eficiencia en el tiempo de recogida de datos de los sensores a nivel de planta. La exactitud del modelo diseñado para sensores se mejora mediante una realimentación de los datos proporcionados por los controladores en tiempo real. Sin embargo, no se evalúa el uso de este middleware con redes que se utilizan actualmente en la industria, por lo que su aplicabilidad en comunicaciones industriales a corto plazo no sería factible.

Lee et al. [94], [95] presentan trabajos relacionados con la aplicación de CPPSs en la automatización industrial en donde proporciona una revisión del Estado de Arte de los CPPSs, y valoran el impacto en la informática industrial de nuevas tecnologías como procesamiento inteligente de datos (Big Data), computación en la nube (Cloud Computing), etc. Como conclusión, la aplicación del CPPS y el análisis inteligente de grandes volúmenes de datos es una solución factible para proporcionar predicción y eficiencia en la personalización de productos, y mejorar la producción en una planta industrial. Sin embargo, no se abordan problemáticas como la reutilización de software, comunicaciones de planta o su uso en plataformas hardware para CPPS.

Actualmente existen aplicaciones CPPS en empresas industriales reales. Uno de los principales retos es diseñar diferentes metodologías de modelado que permitan manejar sistemas de producción industrial complejos y donde se presume que su complejidad au-

mentará a futuro. En este sentido, para sistemas de automatización distribuida con procesos físicos integrados son necesarias metodologías de diseño basadas en técnicas de modelado como las que se pueden soportar bajo la norma IEC 61499.

Desde la perspectiva del diseño, se debe tener modelos de software para las diversas plataformas existentes, modelos para la semántica de ejecución y modelos para las comunicaciones usando redes de comunicaciones industriales. Lo anteriormente expuesto se indica en el trabajo realizado por Dai et al [96] en donde el principal objetivo es presentar modelos integrados de control y software de comunicación para diseños de sistemas distribuidos en aplicaciones Ciber-físicas usando la norma IEC 61499. Sin embargo, el diseño para nuevas normas de comunicación como OPC-UA no se encuentra cubierto en dicha investigación.

Ferrarini et al. [97], [98] promueve un modelado de procesos industriales bajo la norma IEC 61499 basado en la utilización de componentes de automatización industrial con una distribución similar a la estructura de planta o de la maquinaria de la división de procesos dentro de las fábricas. En esta misma línea, Thramboulidis [99] aplica técnicas de modelado usando la norma IEC 61499 y presenta la arquitectura denominada MIM la cual está basada y adapta conceptos de MDA (Model Driven Architecture) planteando un entorno para modelar componentes mecatrónicos que intervienen en procesos industriales.

Christensen [100] propone patrones de modelado para diseño de sistemas industriales distribuidos como apoyo a la norma IEC 61499. De la manera similar, Serna et al. [19] proponen dos patrones de diseño para el manejo de los estados en FBs cuando existen errores en aplicaciones dentro de la norma IEC 61499.

En resumen, estas investigaciones se basan en la coincidencia de las técnicas de búsqueda de nuevas metodologías para producir sistemas distribuidos bajo la norma IEC 61499, pero no se centran en la integración con un protocolo de comunicación útil en fábricas inteligentes, tal como OPC-UA podría ser.

2.9 CONCLUSIONES

El objetivo en este capítulo ha sido analizar brevemente la evolución de las aplicaciones y normas de automatización industrial en la que se refleja claramente que la estandarización ha sido un factor clave para el progreso de la automática a nivel industrial, pero que necesitan ser actualizadas ya que los requisitos adicionales que piden los nuevos conceptos en automatización como son la Industria 4.0, IoT, etc no podrán satisfacerlos, es aquí donde la norma IEC 61499 y OPC-UA pueden intervenir para realizar automatización de acuerdo a las necesidades actuales.

El análisis de trabajos relacionados permite concluir que existe interés en las industrias en aplicar OPC-UA como mecanismo para transmitir datos a nivel industrial. Esta norma proporciona muchas características importantes para los sistemas de control actuales como son: modelado de datos, pilas de comunicación para sistemas empotrados y transmisión segura de datos. Modelos adicionales de información dados por compañías específicas pueden ser añadidos al modelo de OPC-UA y se pueden utilizar para estandarizar modelos entre diferentes proveedores de un mismo campo. Esta tesis demuestra que es factible usar en CPPS de bajo coste y mecanismos de transmisión basados en OPC-UA para el intercambio de información en industrias actuales y futuras.

Los trabajos analizados fundamentalmente tratan de aplicar el paradigma de la Industria 4.0 y comunicaciones M2M mediante el uso de la norma IEC 61499 en las aplicaciones de automatización industrial. Pero, o bien son aproximaciones cerradas que no aprovechan la potencialidad de las herramientas comerciales de fabricante, o bien sustituyen a ésta siendo dificultosa la puesta en marcha de los sistemas, ya que en ambos casos es necesaria en última instancia, una traducción manual.

Asimismo, se puede determinar que queda evidente la necesidad de establecer metodologías que permitan afrontar el diseño de sistemas complejos y distribuidos en la industria de procesos. Ya que, en última instancia, los sistemas de control industrial son sistemas software, también se ha hecho una revisión de la evolución del diseño de sistemas software a nivel industrial y cuáles son sus carencias al momento de aceptar los nuevos paradigmas de automatización que Industria 4.0 e IoT brinda. El análisis ha permitido

constatar el paso de un diseño centrado en la orientación a objeto a un diseño centrado en el concepto de modelo, especialmente en el modelo de las comunicaciones industriales.

En los siguientes capítulos de la tesis se presenta una aproximación que pretende utilizar las metodologías y tecnologías de la Ingeniería de Software analizadas en este capítulo, así resultados de otros trabajos en el campo de la automatización industrial. El diseño basado en modelos y la definición de niveles de abstracción con objeto de aislar al usuario de las tecnologías subyacentes son, entre otras, tecnologías que permiten plantear un entorno de código abierto e independiente de herramienta que dé soporte al ciclo de desarrollo.



3

METODOLOGÍAS PARA DISEÑO DE CPPS

Es evidente que a futuro gran parte de las tecnologías recogidas bajo el marco de la denominada Industria 4.0 tendrá un profundo impacto en todas las empresas y, entre ellas, las relacionadas con la explotación y producción de petróleo y gas. Sin embargo, la apresurada adopción de tecnologías de la información sin un conocimiento adecuado de las mismas y su aplicación a los diferentes aspectos de los sistemas de control de planta, sistema de planificación de recursos empresariales (Enterprise Resource Planning – ERP) y sistemas orientados a la ejecución de la fabricación (Manufacturing Execution Systems – MES) puede derivar en que las compañías de petróleo y gas automaticen sus sistemas industriales con un injustificado alto coste, de manera potencialmente peligrosa y propensa a errores.

La automatización de bajo coste promueve arquitecturas de referencia rentables y nuevos enfoques de desarrollo para aumentar la flexibilidad y la eficiencia de las operaciones de producción en una planta industrial. Esto ha llevado a la adopción por parte de

las empresas de estándares de redes industriales para las comunicaciones a todos los niveles.

En este sentido, OPC-UA, además de permitir adherirse a las empresas a la iniciativa de la Industria 4.0, proporciona acceso local y remoto a la información de planta, facilitando un mecanismo reconocido de integración tanto horizontal como vertical de manera correcta, segura y eficiente. El objetivo principal de este capítulo es presentar una arquitectura abierta para la integración vertical basada en sistemas Ciber-Físicos de producción, configurados bajo la norma IEC 61499 y usando OPC-UA, apta para su utilización en la fabricación flexible en la industria de petróleo y gas. Un segundo objetivo es diseñar metodologías para generación de CPPs desde su introducción y desarrollo, hasta su aplicación.

En este capítulo, primero se introduce una arquitectura para integración de CPPSs en la industria, posteriormente se explica los elementos básicos de los SIFBs y su importancia. A continuación, se presenta una propuesta de metodología de diseño de CPPS. La metodología se ilustra en un entorno de desarrollo, mediante la creación de un repositorio de SIFBs de comunicaciones reutilizables basados en OPC-UA. Estas normas han sido adoptadas por todas las ramas a nivel industrial ya que aborda y resuelven los siguientes problemas: la falta de un modelo universal para el control industrial, la dificultad en la comunicación de las necesidades del usuario, la integración entre proveedores de automatización de proceso y la dificultad en la configuración de control. Para finalizar este capítulo se presenta un caso de estudio donde se usarán los SIFBs diseñados y programados para resolver un problema específico en la industria petrolera ecuatoriana.

3.1 INTRODUCCIÓN

El Internet Industrial de las Cosas (Industrial Internet of Things - IIoT) define un conjunto de tecnologías para el acceso remoto al sistema productivo[55]. Además, el Internet de Servicios (Internet of Services - IoS) [101] proporciona mecanismos aptos para ser empleados para la fabricación basada en la nube y virtualización de procesos. Estas

tecnologías tendrán un impacto colectivo en todos los aspectos de las empresas manufactureras y se hace necesario introducir estos avances en industrias reales como es la del sector de petróleo y gas. [7]

Por otra parte, la norma de automatización IEC 61499 [102] promueve un marco para el desarrollo basado en modelos para sistemas de control distribuidos. La Industria 4.0 en unión con IEC 61499 permitirá modelar y desarrollar componentes software y hardware para sistemas de control distribuidos de manera independiente. La entidad clave en la norma IEC 61499 es el Bloque de Función (Function Block - FB), que encapsula en un mismo elemento los algoritmos de control y comunicación desarrollados en diferentes lenguajes de programación, incluyendo IEC 61131, Java, C ++ o prácticamente cualquier otro tipo de lenguaje. [4]

El Bloque de Función de Interfaz de Servicio (Service Interface Function Block - SIFB) es uno de los tipos de FBs proporcionados por la norma, que permite encapsular y abstraer al usuario del acceso al hardware, comunicaciones, así como a los recursos de la Interfaz de Programación de Aplicaciones (Application Program Interface - API). En resumen, la tecnología actualmente disponible es lo suficientemente madura como para lograr el objetivo de la comunicación M2M, pero todavía hay una brecha entre la tecnología y las industrias reales.

El uso de estándares como IEC 61499 y arquitecturas basadas en CPPS implicaría un gran avance en las tecnologías de la información permitiendo que la industria de petróleo y gas, en su faceta de integración de comunicaciones a nivel de planta, sea eficaz y eficiente. Esta es la razón por la cual se están llevando a cabo numerosas e intensas investigaciones sobre tecnología de la información en los últimos tiempos y éstas están teniendo un impacto inimaginable en este tipo de industria.

El presente capítulo, contribuye a la adaptación de esta arquitectura genérica al caso particular de la industria de producción de petróleo. Esta nueva propuesta de arquitectura basada en CPPS de campo permite la recogida y tratamiento de datos de producción, utilizando el estándar IEC 61499 y el protocolo OPC-UA. Además, sirve como fuente de

información tanto a equipos consumidores de información como a equipos y aplicaciones de nivel superior.

3.2 TRABAJOS RELACIONADOS

En el dominio de la automatización industrial, el enfoque de control más utilizado es la norma IEC 61131-3, que se ha desarrollado para proporcionar un modelo común para la programación de PLCs [103]. Su objetivo principal se enfoca en sistemas de control industrial, los cuales están formados por uno o más controladores acoplados por redes industriales dentro de un área geográfica pequeña, lo que significa que los datos compartidos entre los dispositivos se replican mediante un servicio de comunicación externo a esta norma.

Por el contrario, la IEC 61499 proporciona un enfoque de automatización que se ha desarrollado para modelar sistemas de medición y control de procesos industriales distribuidos con un enfoque principal en el soporte de múltiples proveedores. Esta norma define una arquitectura de referencia para sistemas de automatización y control distribuidos utilizados en entornos industriales (sistemas de fabricación, sistemas de energía y energía, automatización de edificios, etc.). A diferencia de otros estándares de automatización como IEC 61131, el enfoque de IEC 61499 se centra en toda la solución de control; cubre el modelado de las aplicaciones de control, proporciona interfaces abstractas para la comunicación y la interacción de procesos, pero también describe brevemente la configuración subyacente del hardware distribuido. [104], [105]

La integración de diferentes protocolos industriales en la norma IEC 61499 usando SIFBs ya ha sido demostrada por varios grupos de investigación [106]–[108]. Todos ellos tienen en común que se dirigen principalmente a aplicaciones de automatización industrial. Hasta la fecha, las aplicaciones relacionadas con la integración inteligente de comunicaciones industriales se han centrado principalmente en el enfoque de interoperabilidad con la norma IEC 61850. [109], [110]

Los enfoques de comunicación modernos, como las arquitecturas orientadas a servicios y los servicios web, son comunes en las aplicaciones de Internet y en la nube.

Actualmente reciben una creciente atención para la realización de sistemas de red inteligente [111]. En este dominio, el "Perfil de Dispositivos para Servicios Web" (DPWS), así como la Transferencia de Estado Representacional (del inglés Representational State Transfer REST), basada en SOAP, proporcionan enfoques importantes cuando se implementan en controladores embebidos de nueva generación [53]. DPWS se ocupa principalmente de la omnipresente integración de dispositivos con servicios web en dispositivos de control distribuido, mientras que REST puede ser visto como un enfoque de integración alternativo que surgió de la iniciativa de World Wide Web y se centra principalmente en la comunicación M2M.

Tradicionalmente, los problemas relacionados con la utilización de CPPS para automatización industrial se han abordado por separado tanto en la ciencia de la computación como en la del control industrial, pero rara vez en una conjunción sinérgica. Esto es insuficiente en vista de que el concepto de IIoT se convierte en un motor importante para varias aplicaciones industriales. En la fabricación industrial, conduce a la reducción de los niveles jerárquicos de la pirámide de automatización aumentando así la eficiencia y permitiendo un nivel sin precedentes de flexibilidad de producción y adaptabilidad. Esto hace que sea factible producir productos en cantidades menores con un tiempo más corto a los mercados y una mayor eficiencia económica y estos conceptos pueden ser usados en optimizar los procesos de la industria petrolera ecuatoriana.

De acuerdo con los conceptos de Industria 4.0 [112], las principales fuerzas motrices de la nueva revolución industrial son IIoT y CPPS. En el entorno de fabricación, CPPS comprende máquinas inteligentes, sistemas de almacenamiento e instalaciones de producción capaces de acciones autónomas y de colaboración. El creciente interés en el uso de arquitecturas de hardware distribuidas en la automatización ha llevado a la integración de CPPPs con la norma IEC 61499, que presenta una arquitectura software basada en componentes. IEC 61499 posee una arquitectura que utiliza el concepto de invocación por eventos de componentes que se correlaciona bien con el mecanismo de transmisión de mensajes de la comunicación en red. Esto ayuda a lograr cierto grado de transparencia en la asignación de software al hardware usado en automatización industrial.

La teoría de los sistemas de control ha reaccionado a la revolución de la Industria 4.0 debido a la gran cantidad de trabajos sobre control en redes distribuidas y, específicamente, con la implementación de teorías de control basado en eventos, desarrolladas por Åström [113] y Årzen [114]. Los métodos de control en redes de comunicaciones se basan en robustos algoritmos de control con el propósito de reducir el impacto de la variabilidad del período de muestreo sobre la calidad del control. Tales métodos de control avanzados ayudan a lograr una mejor calidad de control en el caso de tasas de muestreo inciertas. Sin embargo, son más complejos en la implementación y tienen un mayor consumo de recursos computacionales, por lo que es de difícil implementación en CPPS de bajo coste.

El uso eficiente de métodos impulsados por eventos por parte de los desarrolladores de sistemas de automatización se ve obstaculizado por la falta de soporte en lenguajes de programación, dispositivos de control comercial y herramientas de diseño. Las investigaciones en el ámbito de la informática se han preocupado más por investigar y garantizar las propiedades del tiempo de ejecución del software en un único nodo informático o en una red distribuida. Este es un requisito previo importante, pero no lo suficiente para cumplir con los requisitos de rendimiento de los sistemas físicos que realizan el control industrial y es por esta razón que el uso de las arquitecturas dadas por IEC 61499 serían de gran ayuda para resolver estos problemas.

El término CPPS ha sido acuñado para enfatizar la naturaleza interdisciplinaria de las interacciones informáticas con el mundo físico, en el que ambas partes se influyen mutuamente y el diseño eficiente es imposible mantenerse dentro de las disciplinas clásicas de la informática, la teoría del control o las comunicaciones. Modelar CPPS con el fin de validar su comportamiento es un problema intrínsecamente difícil debido a la necesidad de tener en cuenta aspectos de corrección computacional, aspectos de corrección de control, comportamiento dinámico del entorno físico junto con la comunicación y especialmente las dependencias cruzadas entre estos factores.

En lo que respecta al diseño de sistemas de automatización distribuidos con una verdadera lógica distribuida, la arquitectura de FBs introducida en la IEC 61499 sigue siendo la arquitectura de referencia de nivel de sistema más creíble con perspectivas de

adopción industrial suficientes y potencial para definir una semántica formal adecuada. Reconociendo esta tendencia tecnológica, ha habido un creciente interés en el dominio de la automatización industrial en aplicaciones eficientes de esta tecnología para obtener beneficios tangibles en el diseño de sistemas distribuidos confiables.

Por ejemplo, se han llevado a cabo varias investigaciones para probar la viabilidad del uso de IEC 61499 con sistemas de automatización basados en actuadores con capacidad de comunicación punto a punto [115], [116]. Las arquitecturas multi-agentes se han aplicado a nivel de la capa de aplicación para lograr la autonomía y la autoconfiguración de los sistemas, en particular en la automatización eléctrica usando la metodología de las redes eléctricas inteligentes (del vocablo inglés SmartGrid)[57], [110], [117]. La arquitectura orientada al servicio (SOA) se ha ampliado y superado con las actuales prácticas de automatización [49], [53], [118] para permitir la interoperabilidad, simplificación del diseño y la arquitectura de componentes cibernéticos (CPCA), fue introducido en [119] para facilitar el diseño y validación de aplicaciones en el entorno IIoT.

Sin embargo, hasta el momento, los diferentes trabajos sobre la norma IEC 61499 se han centrado en los procesos de control discreto. En este capítulo se da un paso adelante y se presenta la utilización de la norma IEC 61499 para la monitorización de procesos analógicos realizando una integración de planta empleando protocolos de comunicaciones industriales.

Investigadores se han enfocado en el diseño e implementación de CPPS de bajo coste bajo la norma IEC 61499 [120]–[122]. Sin embargo, no ha sido tratada la integración entre servidores OPC-UA y protocolos de comunicación industrial propietarios. El servidor OPC-UA propuesto en este capítulo está concebido para un sistema de automatización de una industria real como es el petróleo y gas y modelado sobre un entorno XML.

3.3 INTEGRACIÓN VERTICAL

Industria 4.0 permite y amplía el intercambio automatizado de información entre la producción industrial y los procesos de negocio con el objetivo de reducir costos y

umentar la producción. Para lograr esto, la interoperabilidad es necesaria para garantizar la facilidad de uso global y la coherencia entre sistemas de producción industrial. Esta sección se centra en el eje de integración vertical y presenta los requisitos identificados, así como una selección de métodos, técnicas y tecnologías actualmente disponibles, que se utilizan ampliamente en sus respectivos dominios.

La integración vertical tiene como objetivo la recopilación de datos del proceso de producción y su transformación en información valiosa para clientes remotos. Para adquirir los datos del proceso, es necesario cumplir dos tipos de requisitos: los que se refieren al *usuario*, es decir, ¿qué datos deben adquirirse? (magnitudes, unidades de ingeniería, rango, etc.) la planta física (¿a qué proceso se refiere?) y la forma en que estos datos tienen que ser capturados (un disparo, de forma síncrona, asincrónicamente, etc.). Por otra parte, desde el punto de vista de la implementación, también son necesarios los medios para describir la planta (¿dónde?), describir los datos a adquirir (¿qué?) y definir el mecanismo de adquisición (¿cómo?).

Para satisfacer las necesidades de los usuarios y la implementación, existen métodos, técnicas y tecnologías que están bien extendidas y consolidadas en la industria y que pueden usarse para cumplir con los desafíos actuales de las empresas. Con respecto a los métodos y técnicas, el paradigma de la Ingeniería Guiada por Modelos (Model-driven engineering) encaja muy bien, ya que ofrece:

- Separación de ambigüedades a través del modelado de dominios, promoviendo descripciones de sistemas desde diferentes puntos de vista.
- Mapeo de modelos de dominio a través de transformaciones de modelo a modelo, soportando análisis de consistencia.
- Generación automática de archivos de configuración y código a través de transformaciones de modelo a texto.

Con respecto a la descripción de la información para los diferentes actores dentro de la automatización industrial, existen estándares consolidados para cada capa en la au-

tomatización de procesos. Algunas normas que permiten esta descripción han sido seleccionados y son presentados a continuación, aunque hay muchos otros estándares que pueden colaborar para lograr la integración real y el intercambio de información, se han seleccionado los de mayor impacto.

a) **Descripción de la topología a nivel de planta:** AutomationML (AML) define modelos que describen las plantas de producción y sus componentes, incluyendo información de topología, geometría, cinemática y lógicas. Actualmente se encuentra aceptado bajo norma IEC 62714. CAEX (IEC 62424) se utiliza para definir un formato de dato neutro para el intercambio de información de plantas entre herramientas CAE heterogéneas. ISO 15926 da un modelo de planta basado en un marco ontológico, provisto de semántica y especificación de tiempo. Por último, la ISO 15531 define un modelo para adquirir datos a nivel de control para ser almacenados en las herramientas de gestión de producción.

AML parece ser muy prometedor ya que define un formato de datos específicamente diseñado para el intercambio de información de ingeniería de planta. El formato permite la integración perfecta de herramientas de ingeniería de diferentes disciplinas y fases del ciclo de vida de un proyecto industrial. En [123] se analiza la interoperabilidad de ambas normas, OPC-UA y AML. Los objetivos son intercambiar las descripciones de los componentes de la planta mediante la creación de los modelos de información OPC-UA basados en AML. Pero no es suficiente para la adquisición en tiempo real de grandes datos del proceso de producción.

b) **Datos de los dispositivos de la planta:** La interfaz PLCopenXML define un modelo para que los PLC sean interoperables mediante el uso de la norma IEC 61131-3. También se utiliza en AML para definir la lógica de la aplicación. Otra norma muy usada es Integración de Dispositivos de Campo (Field Device Integration - FDI) [124], la cual se centra en la integración de dispositivos para aplicaciones de automatización de procesos, basadas en las tecnologías de Lenguajes de Descripción de Dispositivos (Electronic Device Description Language – EDDL) y Gestor de Tipos de Dispositivo (Device Type Manager - DTM). Finalmente, la Red de

Sensores Semánticos (Semantic Sensor Network - SSN) [125] define modelos para sensores incluyendo descripción, precisión, capacidades, observaciones y métodos utilizados para el sensor.

c) **Intercambio de información:** La norma IEC 61850 [117] define un conjunto de modelos para dar acceso a subestaciones eléctricas mientras se ha definido una extensión, la norma IEC 61400-25 [126], para plantas eólicas. Es una norma madura y bien establecida que posee modelos de información que representan conceptos, relaciones, restricciones, reglas y operaciones para especificar la semántica de los datos. Aunque la automatización es mucho más amplia en términos de conceptos, dispositivos y tipos de datos y equipos, el concepto de definir modelos adaptados podría ser útil en la fabricación.

d) **Tecnologías de la comunicación:** OPC-UA (IEC 62541) ofrece mecanismos de middleware para integrar plataformas heterogéneas. El modelo de información utiliza conceptos de alto nivel, como Mahnke et al. han mostrado [127], así como también poseen, entidades con jerarquías, agregación, variables y tipos de datos. La norma de comunicación Time Sensitive Networking (TSN) es un protocolo determinista en tiempo real para Ethernet que puede sincronizar dispositivos con precisión en la red, eliminando la necesidad de los métodos de sincronización basada en señales que son comunes en la actualidad. Este concepto distribuido de tiempo también permite la programación de tráfico y la transferencia determinística de datos sensibles, lo cual es clave para aplicaciones industriales como procesos y control de máquinas donde es crítico tener baja latencia y mínima fluctuación para cumplir con los requerimientos de control de ciclo cerrado.

3.4 ARQUITECTURA CPPS

La arquitectura preliminar de este trabajo la presentó Pérez en [12] y es tomada como un punto de partida para implementarla usando la norma IEC 61499 en unión con el protocolo OPC-UA. La Figura 23 representa el escenario general para el proceso de monitorización empleando dispositivos CPPS. Como se ha comentado anteriormente, esta arquitectura se centra en la integración vertical entre los procesos productivos y las

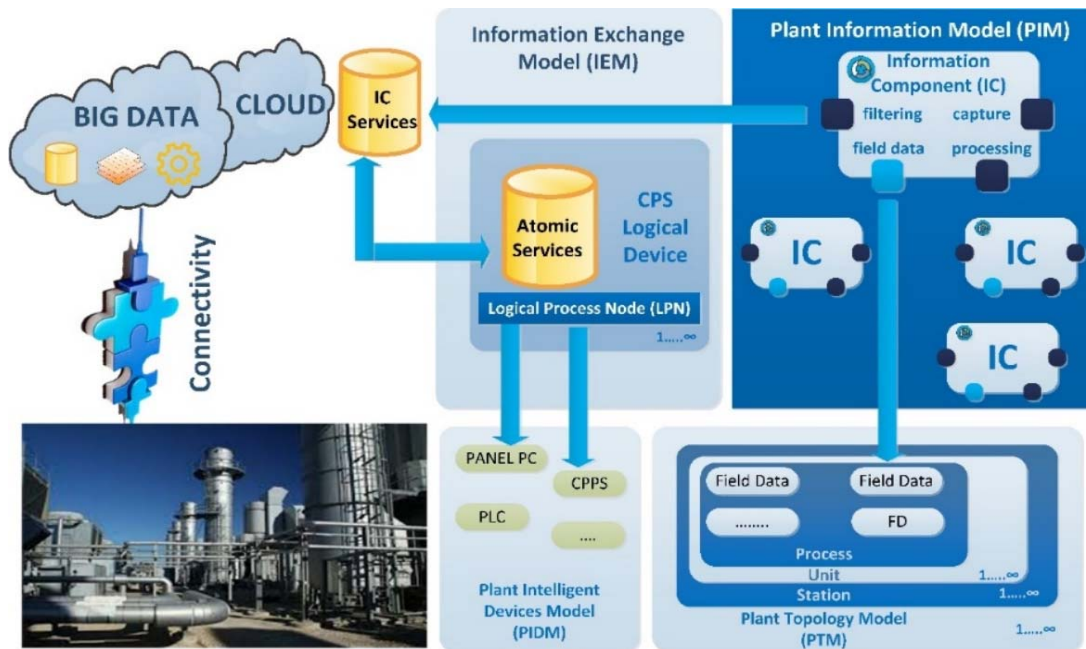


Figura 23: Arquitectura General para dispositivos CPPS

áreas de negocio. Por lo tanto, el objetivo es dar acceso a los datos del proceso de producción y extraer la información a través de análisis de gran cantidad de datos (Big-Data analysis).

Esta arquitectura se compone de un conjunto de componentes que gestionan un conjunto de modelos que representan el mundo físico, el intercambio de información, así como la información que debe ser recogida de la planta. En los siguientes párrafos se detallan los conceptos manejados por cada componente de la arquitectura.

3.4.1 Modelo de Procesos de Producción

El mundo físico, entendido como el proceso de producción, se denomina Modelo de Procesos de Producción (en inglés Production Process Model o PPM). Este modelo está compuesto por los datos accesibles, generados en el proceso de producción (fuente de datos), y también de los dispositivos inteligentes pertenecientes a la planta que son responsables de transmitir los valores de los datos en tiempo real (proveedores de datos). Está formado por las siguientes partes:

3.4.1.1 Modelo de Topología de Planta

La planta física, incluyendo las variables del proceso, se define mediante el denominado Modelo de Topología de Planta (en inglés Plant Topology Model o PTM). Este modelo recoge los componentes físicos de la planta. El PTM representa la disposición de los componentes en la planta. Debe ser suficientemente genérico para describir cualquier proceso productivo, aunque puede personalizarse para ajustarse al interés de la empresa.

Por ejemplo, la jerarquía: Estación → Unidad → Proceso → Datos de campo, puede ser usada para representar una planta compuesta por un número de estaciones. Un conjunto de unidades puede componer una estación y una unidad puede tener un conjunto de procesos (fuentes de datos) asociados. Independientemente del número de capas de la jerarquía, en la capa inferior se definen el conjunto de procesos caracterizados por sus propiedades de datos (magnitud física, unidades de ingeniería, rango de posibles valores o marca de tiempo, entre otros).

3.4.1.2 Modelo de Dispositivos Inteligentes de Planta

Los dispositivos proveedores de datos, que son controladores de procesos y/o dispositivos inteligentes que ofrecen datos de proceso y sus características, se definen en el denominado Modelo de Dispositivos Inteligentes de Planta (en inglés Plant Intelligent Devices Model o PIDM). Adicionalmente, se tiene el Modelo de Información de Planta (en inglés Plant Information Model o PIM), corresponde a los requisitos de usuario para monitorización que pueden definirse a partir del PTM especificando las variables de proceso, así como el tipo de acceso.

3.4.2 Modelo de Intercambio de Información

Otro modelo importante es el Modelo de Intercambio de Información (en inglés Information Exchange Model o IEM), que procesa los requisitos de usuario para la supervisión y es el responsable del acceso a los datos apropiados de los dispositivos del proceso industrial. El IEM se compone de tres entidades diferentes:

3.4.2.1 Servicios Atómicos

Los Servicios Atómicos (del inglés Atomic Services o AS) son un conjunto de servicios básicos que realizan el acceso a los datos de cualquier proceso. Existen varios tipos de AS que cubren diferentes aspectos como la seguridad, filtrado, pre-procesamiento de instrucciones para mando y monitorización de datos, por ejemplo: Obtener datos / establecer valores a datos, publicar/suscribir, reportar, registrar datos de campo, etc.

3.4.2.2 Nodos Lógicos de Proceso

Los Nodos Lógicos de Proceso (en inglés Logical Process Nodes o LPN), representa procesos del PPM. Existen un LPN por proceso y cada LPN hace referencia al dispositivo inteligente donde se despliegan los servicios para acceder a los datos del proceso. Por lo tanto, cada proceso definido en el PPM está relacionado con un dispositivo inteligente que es responsable de proporcionar el conjunto de datos relacionado al proceso industrial.

3.4.2.3 Dispositivos Lógicos CPS

Los dispositivos Lógicos CPS (en inglés CPS Logical Devices o CPSLD), son los dispositivos inteligentes del PPM donde se despliega la arquitectura del proceso. CPPs gestiona los modelos del PTM y PIDM y es el que permite implementar todas las funcionalidades del modelo IEM. Además, cada CPSLD por lo menos posee un LPN del proceso industrial.

3.4.3 Modelo de Información de Planta

Finalmente, el modelo PIM define la información del proceso accesible por el usuario. Esta información está estructurada en los llamados Componentes de Información (en inglés Information Components o IC). Los IC son los mecanismos que permiten acceder a información generada a partir de datos de planta. Están compuestos a partir de un conjunto de servicios atómicos proporcionados por los Nodos Lógicos de los CPS (CPSLD) los cuales manejan los datos de campo del proceso (en inglés Field Device o FD). El usuario define los ICs mediante servicios de configuración. Una vez creado e iniciado, un

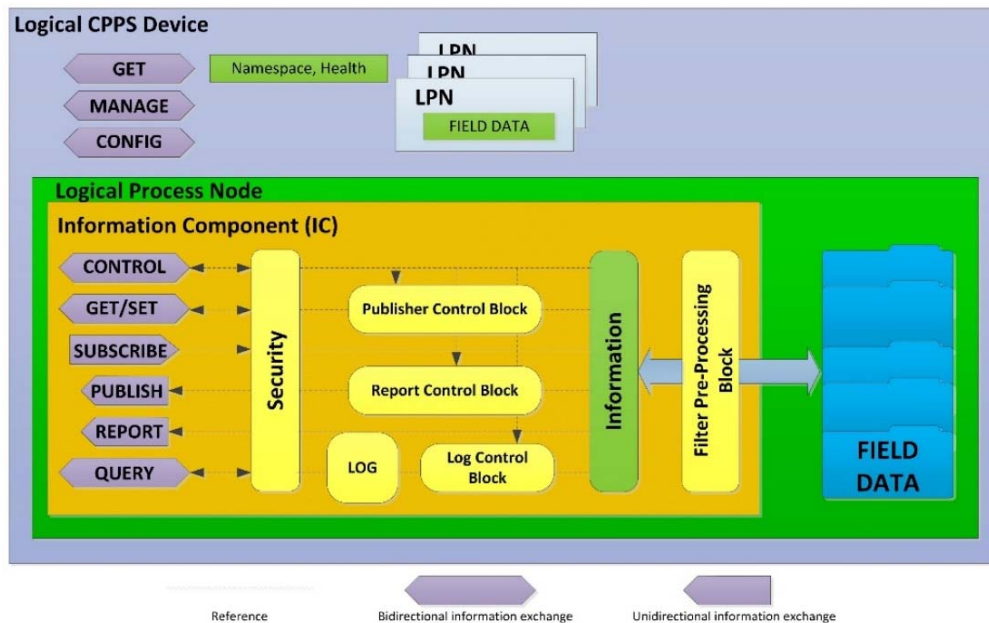


Figura 24: Estructura conceptual del IEM

IC ofrece un único servicio (en inglés Information Component Services o ICS) que es responsable de implementar la funcionalidad del IC. El IC más simple es el acceso mínimo a la planta, por ejemplo, el acceso de lectura a datos de campo de un proceso (que está asociado a un LPN). Este es un modelo orientado al usuario, por lo que las aplicaciones cliente remotas son responsables de definir los servicios. La Figura 24 muestra la estructura conceptual del Modelo de Intercambio de Información donde se indican diferentes funcionalidades de los servicios atómicos para adquisición y pre-procesamiento, y como se refieren a datos de campo pertenecientes a procesos recogidos en el Modelo de Topología de Planta.

3.5 IEC 61499 COMO HERRAMIENTA PARA INTEGRACIÓN DE ARQUITECTURAS CPPS

La arquitectura basada en redes de FBs para dispositivos de control permite un enfoque de diseño estructurado modular, lo que permite un proceso de desarrollo sea más simple y eficiente. Esta norma permitiría implementar la arquitectura CPPS explicada en el apartado anterior. Sin embargo, dado que la norma IEC 61499 presenta un modelo de

referencia conceptual genérico, si se desea una perspectiva de aplicación orientada a objetos es necesario establecer un modelo derivado para la clase FB. Por consiguiente, en este apartado se explicará el modelo y los elementos básicos de un Bloque de Función de Interfaz de Servicio (SIFB) ya que es el que permitirá implementar físicamente lo teorizado en la arquitectura explicada.

El Bloque Funcional de Interfaz de Servicio SIFB, es un bloque funcional que proporciona servicios a la aplicación, permitiendo la interacción entre aplicación y recursos. Proporciona una interfaz entre el software de control y el hardware o los sistemas de comunicaciones. El estándar IEC 61499-1 dentro de su especificación de Bloque Funcional, define tres clases de bloques funcionales:

1. **Bloque de Función Básico:** En este tipo de FB el control de ejecución está especificado por una Gráfica de Control de Ejecución (ECC, Execution Control Chart). Los algoritmos que van a ser ejecutados son declarados como se especifica en los perfiles de conformidad en el Apartado 4 del estándar IEC 61499.
2. **Bloque de Función Compuesto:** Los algoritmos y el control de ejecución son especificados a través de conexiones de eventos y de datos en uno o más FBs (redes de FBs).
3. **Bloque de Función de subaplicación:** En este tipo de FB, los algoritmos y la ECC son especificados por los tipos de FBs compuestos, pero con la propiedad específica que el componente FB o las subaplicaciones puedan ser distribuidos entre varios recursos.

Además, también define una clase especial de FB, el SIFB. Es un bloque funcional que proporciona uno o más servicios a una aplicación, basado en el mapeo de primitivas de servicio a la entrada de eventos, salida de eventos, entrada de datos y salida de datos del bloque funcional.

3.5.1 Elementos básicos de un SIFB

Los elementos básicos de un SIFB los resume el estándar IEC 61499-1. Estos elementos básicos se definen en una plantilla general de un SIFB como se muestra en la

Figura 25. Sin embargo, una de las funciones principales del SIFB es proporcionar servicios y el estándar IEC 61499-1 define los siguientes conceptos:

- **Servicio:** Es la capacidad funcional de un recurso, la cual puede ser modelada por una secuencia de primitivas.
- **Primitivas:** Es una representación abstracta de una interacción entre una aplicación y un recurso.

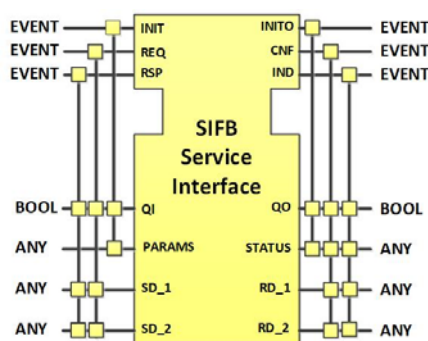


Figura 25: Estructura General de un SIFB

Como se ha mencionado en las definiciones del SIFB, la relación con los servicios es importante, ya que se utiliza en la descripción de los elementos del SIFB. Para declarar los tipos de SIFBs, el estándar define el conjunto de entradas de eventos, salidas de eventos, entradas de datos y salidas de datos, listados en las Tablas 4 y 5.

Tabla 4: Eventos y datos de entrada estándar de un SIFB

Entrada Evento	Entrada de Dato
<p>INIT Esta entrada de evento debe ser asignada a una primitiva de petición la cuál solicita una inicialización del servicio proporcionado por la instancia del FB.</p>	<p>QI : BOOL Esta entrada representa un calificador en la primitiva de servicio asignada a la entrada de evento. Si esta entrada es VERDADERA sobre la ocurrencia de un evento INIT, el servicio de inicialización es solicitado; si es FALSA, el servicio de finalización es requerido.</p>
<p>REQ Esta entrada de evento debe ser asignada a una primitiva de petición del servicio proporcionado por la instancia del FB.</p>	<p>PARAMS : ANY La entrada contiene uno o más parámetros asociados con el servicio, típicamente como elementos de una instancia de un tipo de dato estructurado. Cuando esta entrada está presente, la especificación del tipo de FB debe definir el tipo de dato y valor inicial por defecto.</p>

RSP Esta entrada de evento debe ser asignada a una primitiva de respuesta del servicio proporcionado por la instancia del FB.	SD₁,..., SD_m : ANY Estas entradas contienen el dato asociado con las primitivas de petición y de respuesta. La especificación del tipo FB debe definir los tipos de datos y valores por defecto de estas estradas y debe definir sus asociaciones con entradas de eventos en un diagrama de secuencia de eventos.
---	---

Tabla 5: Eventos y datos de salida estándar de un SIFB

Salida Evento	Salida de Dato
INITO Esta salida de evento debe ser asignada a una primitiva de confirmación que indica la finalización del procedimiento de un servicio de inicialización.	QO : BOOL Esta variable representa un calificador en la primitiva de servicio asignada a la salida de eventos. Por ejemplo, si el valor de esta salida es VERDADERO sobre la ocurrencia de un evento INITO indica la inicialización exitosa del servicio; si esta salida tiene un valor FALSO indica la inicialización NO exitosa.
CNF Esta salida de evento debe ser asignada a una primitiva de confirmación del servicio proporcionado por la instancia del FB.	STATUS : ANY Esta salida debe ser un tipo de dato apropiado para expresar el estado del servicio sobre la ocurrencia de una salida de evento.
IND Esta salida de evento debe ser asignada a una primitiva de indicación del servicio proporcionado por la instancia del FB.	RD₁,..., RD_n : ANY Estas salidas contienen los datos asociados con las primitivas indicación y confirmación. La especificación del tipo de FB debe definir los tipos de datos y valores iniciales de estas salidas y debe definir sus asociaciones con salida de eventos en un diagrama de secuencia de eventos.

3.5.2 Especificaciones del SIFB

Las interfaces externas de un SIFB tienen la misma apariencia que un FB. Sin embargo, algunas entradas y salidas de un SIFB tienen semántica especializada y el comportamiento de instancias de estos tipos de SIFBs está definido a través de una notación gráfica especializada para secuencia de primitivas de servicio.

En función del tipo de SIFB, se ofrecen diferentes servicios. El estándar IEC 61499-1 define dos tipos de SIFBs que son representados en la Figura 26.

1. **Petición (REQUESTER)**: Representa una interacción de inicializar aplicación (ejemplo: envío de un mensaje). Este permanece pasivo hasta la llegada de una entrada de evento. Por otro lado, su ejecución es similar a la ejecución de un FB básico.

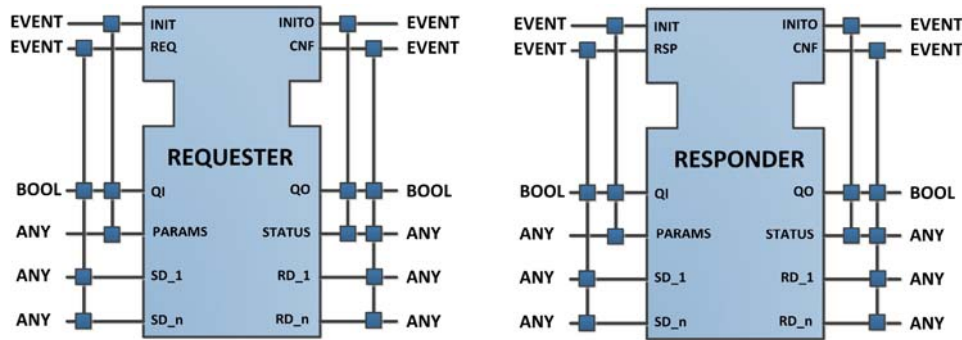


Figura 26: Tipos de Servicio de SIFB Requester y Responder

2. **Respuesta (RESPONDER):** Representa una interacción de inicializar recurso (ejemplo: recepción del mensaje). Si un servicio en reposo lanza su ejecución, este tipo de SIFB proporciona una salida de evento. También puede interrumpir la ejecución de cualquier otro FB.

Las primitivas de servicio definen propiedades de la interacción entre la aplicación y el recurso durante el servicio y por lo tanto tiene otra definición: la primitiva de servicio, es una relación de especificación cualitativa que no especifica los tiempos exactos de requisitos para los servicios.

Algunos servicios para los diagramas de tiempo de secuencia son: establecimiento normal, servicio normal, terminación de la aplicación iniciada. Los SIFBs ejecutan la asignación de la primitiva de servicio a la entrada de eventos de un FB. Cada primitiva de servicio procesa los correspondientes datos de entrada y genera datos y eventos de salida, donde cada primitiva de servicio esta especificada como un diagrama de secuencia.

Tabla 6: Entradas y salidas de un FB de comunicaciones

Variable	Semántica
ID	Esta entrada proporciona los parámetros de la conexión asociada de comunicación a la instancia del FB de comunicaciones. Esto incluirá la forma de identificar el protocolo de comunicación y conexión de la comunicación y puede incluir otros parámetros.
SD_1,..., SD_m	Estas entradas representan los datos que se transfieren por la conexión de comunicación especificada por la entrada ID sobre la ocurrencia de una primitiva REQ+ o un RSP+.

STATUS	Esta salida representa el estado de la conexión de comunicación.
RD_1,..., RD_n	Estas salidas representan los datos recibidos por la conexión de comunicación especificada por la entrada ID sobre la ocurrencia de una primitiva de un IND+ o CNF+.

La declaración de los FBs de comunicaciones es similar a la del SIFB, solo que con semántica especializada en las variables de entrada y salida. Como se muestra en la Tabla 6.

La comunicación es muy importante en los sistemas de control distribuidos. En el estándar IEC 61499-1 se implementa vía SIFBs de comunicaciones. Los dos paradigmas de comunicación empleados son publicista/suscriptor (PUBLISH/SUBSCRIBE) y cliente/servidor (CLIENT/SERVER). La comunicación publicista/suscriptor transmite y recibe información en una única dirección, (ver la Figura 27) es decir, la transferencia de datos es unidireccional.

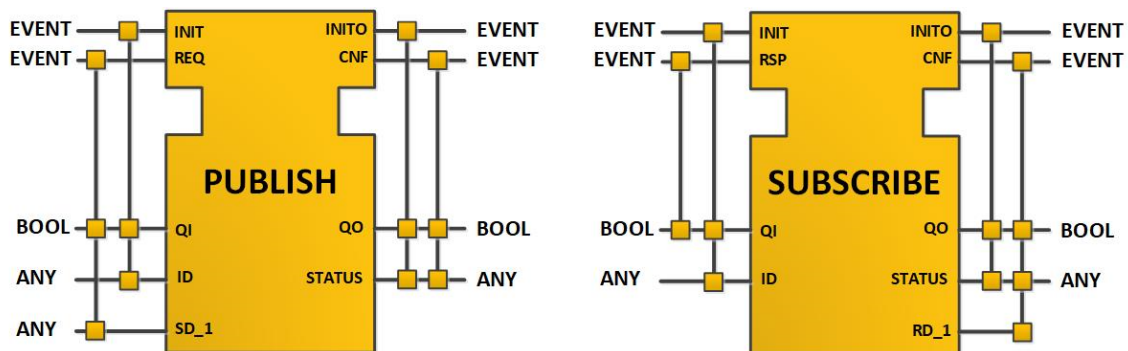


Figura 27: Bloques de interfaz de comunicaciones Publish y Subscribe

El publicista publica la información en la red, mientras que el suscriptor recibe la información y no hay reconocimiento entre los dos. El publicista transmite el valor o los valores de una o más variables a uno o más recursos externos que están suscritos a dichos datos.

Como se ha mostrado en la Figura 27, en el bloque publicista, la línea de la entrada de evento INIT inicializa el bloque. La barra vertical asociada con la línea, llamada construcción “WITH” muestra qué línea de datos se lee al mismo tiempo. QI es una línea de control, mientras la línea ID establece los parámetros de la red a través de los que el dato

publicado será comunicado en la red. En el lado de salida, el evento INITO indica que la inicialización está completa y las líneas de salida QO y STATUS muestran el resultado de esa inicialización.

El enlace entre publicista y suscriptor se lleva a cabo a través de la conexión ID. Una vez inicializado, el publicista indica presencia con la conexión ID, enviando eventos para notificar al suscriptor. Cuando el suscriptor sea inicializado, podrá escuchar por la conexión ID y los parámetros de comunicación requeridos. El suscriptor notifica para establecer la conexión con el publicista y le solicita los datos.

La comunicación cliente/servidor transmite y recibe información en dos direcciones, (ver Figura 28), es decir, la transferencia de datos es bidireccional. El cliente usualmente envía la petición de servicio y el servidor recibe y responde a estos. Cada mensaje lleva su reconocimiento mediante un protocolo punto a punto.

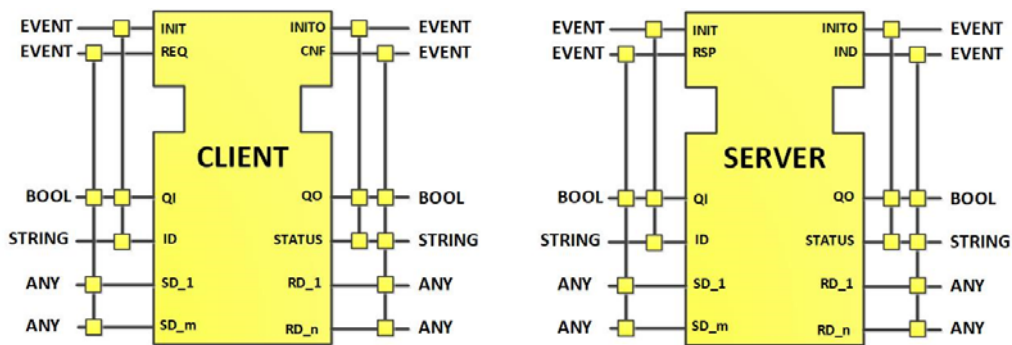


Figura 28: Bloques de interfaz de comunicaciones Client y Server

Dentro del modelo cliente/servidor un sistema de software distribuido se compone de una tarea servidora a una tarea cliente. La tarea cliente envía una petición (Request) al servidor de acuerdo al protocolo definido el cual es procesado por el servidor. Dependiendo de la acción o información solicitada por el cliente, el servidor envía una respuesta (Response) de regreso al cliente.

Como hemos señalado antes, es evidente que los SIFBs son importantes en el diseño de un sistema de control distribuido, pero tal como lo menciona el estándar, las comunicaciones son abiertas, es decir, los desarrolladores pueden utilizarlas como mejor les convenga para resolver sus aplicaciones de control distribuidas.

Por último, como se mencionó anteriormente, el uso de los SIFBs es muy variado y no existen metodologías de diseño y desarrollo de los SIFBs. Además, en los sistemas de desarrollo actuales que permiten crear aplicaciones de control distribuidas y cumplen con el estándar IEC 61499, no existen tipos de guías para la implementación de SIFBs y hasta este momento los SIFBs se han introducido manualmente.

3.5.3 Metodología de Diseño de SIFB

En el estudio realizado sobre investigaciones referentes a este tema, no se han encontrado planteamientos de otros investigadores relacionados a cómo se diseñan los SIFBs, ya que los presentan como si estuvieran elaborados. Tampoco presentan cómo implementan los SIFBs en los entornos de desarrollo. Además, el estándar solo menciona la descripción general y las especificaciones de los SIFBs.

Por estas razones, una de las aportaciones de esta tesis es una metodología de diseño para crear SIFBs, donde el usuario pueda diseñar, desarrollar y aplicar SIFBs de una manera sencilla. Esta metodología ha sido aplicada usando la herramienta de desarrollo 4DIAC, es una herramienta de software de libre distribución, donde su runtime es FORTE y está implementado en C++.

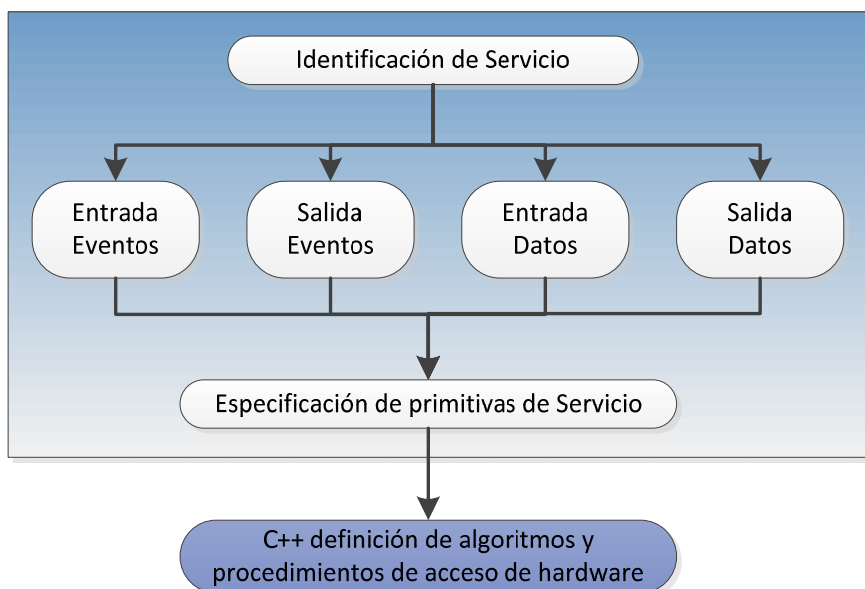


Figura 29: Elementos en el desarrollo de SIFBs en C++

En la Figura 29 se representa la metodología basada en C++. La metodología para desarrollar SIFBs IEC 61499 basados en C++ FORTE, consta de cuatro pasos: Identificar, definir, especificar y desarrollar. Estos pasos se describen a continuación:

1. **Identificar el servicio:** Es necesario identificar el servicio que el SIFB debe implementar. Es uno de los pasos importantes, ya que es el paso previo antes de desarrollar el SIFB. Algunos de los servicios más comunes son: inicializar, comparar, seleccionar, restablecer, separar, activar, leer y escribir (variables de datos o ficheros), etc.
2. **Definir las entradas y salidas del SIFB:** Las definiciones de las entradas de eventos, las salidas de eventos, las entradas de datos y las salidas de datos, son indispensables para proporcionar los servicios. Es importante conocer la cantidad de variables de entrada y salida que serán utilizadas, así como el tipo de variable que puede ser: BOOL, ANY, STRING, WORD, WSTRNG, BYTE, INT, DWORD, REAL, DATE_AND_TIME, etc.
3. **Especificar las primitivas de servicio:** Se tiene en cuenta la especificación de la secuencia de las primitivas de servicio, unidas con los datos asociados para condiciones normal y anormal. Este paso es básico, ya que se especifica cómo se desea que se lleve a cabo el servicio y cómo se debe diseñar la primitiva de servicio.

Después de haber realizado estos tres primeros pasos en el entorno de desarrollo FBDK, se obtiene la estructura general de C++ de un SIFB que se ilustra en la Figura 29. Cada SIFB extiende la clase instancia de FB como un FB especial. La estructura generada incluye los eventos y la gestión de funciones. El diseñador tiene que incluir funciones nativas de usuario y variables de usuario. La estructura de los servicios implementados por el SIFB se declara por la herramienta, pero debe ser modificada por el diseñador con el fin de introducir la funcionalidad deseada. Al final, se definen las funciones de usuario y las funciones auxiliares.

4. **Desarrollar los algoritmos que implementan el servicio:** Definir y desarrollar los algoritmos en C++ y al mismo tiempo incluir las funciones de acceso a hardware.

3.5.4 Implementación de la metodología en 4DIAC

Para llevar a cabo la implementación de la metodología, se han utilizado varias herramientas software previamente seleccionadas. Inicialmente, se ha utilizado la herramienta de ingeniería 4DIAC que emplea la máquina virtual de Java. Esta herramienta permite editar y ejecutar proyectos basados en IEC 61499 bajo el runtime FORTE y el runtime FBRT. Además, se ha utilizado una herramienta de programación C/C++ basada en Eclipse CDT.

La Figura 30 ilustra el escenario general para implementar SIFBs usando las herramientas 4DIAC y Eclipse CDT C/C++. En el desarrollo de los tres primeros pasos de la metodología, se utilizó la herramienta 4DIAC-IDE; primero definir un nuevo SIFB, así como identificar los servicios que el SIFB debe aplicar; segundo definir las entradas de eventos, las entradas de datos, las salidas de eventos y las salidas de datos, necesarias para proporcionar el servicio; tercero especificar las primitivas del servicio a utilizar.

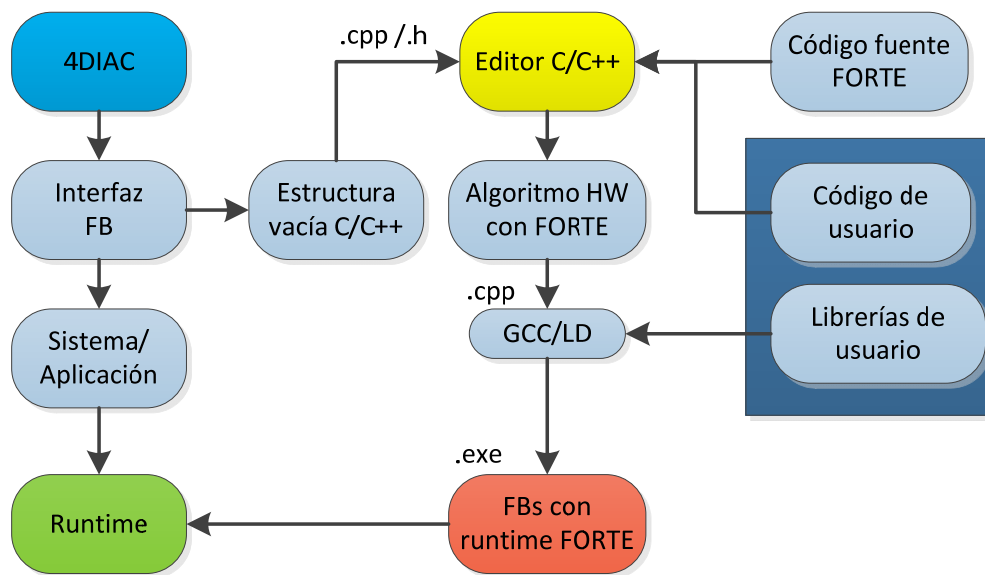


Figura 30: Escenario general de desarrollo del software en 4DIAC

Con estos tres primeros pasos se genera una estructura vacía C++ de un SIFB; en este ejemplo, la estructura del SIFB OPC-UA SERVER.cpp que se ilustra en la Figura 31. La estructura incluye la declaración y tipos de eventos y datos de entrada, eventos y

datos de salida y se añade el acceso al hardware. Esta estructura se exporta como un fichero .cpp y como un fichero .h con estructura FORTE.

```

    #ifndef FORTE_ENABLE_GENERATED_SOURCE_CPP
    #include "OPCUA_SERVER_gen.cpp"
    #endif

    DEFINE_FIRMWARE_FB(FORTE_OPCUA_SERVER, g_nStringIdOPCUA_SERVER)

    const CStringDictionary::TStringId FORTE_OPCUA_SERVER::scm_anDataInputNames[] = {g_nStringIdQI, g_nStringIdCONFIGFILE};
    const CStringDictionary::TStringId FORTE_OPCUA_SERVER::scm_anDataInputTypeIds[] = {g_nStringIdBOOL, g_nStringIdWSTRING};
    const CStringDictionary::TStringId FORTE_OPCUA_SERVER::scm_anDataOutputNames[] = {g_nStringIdQO, g_nStringIdSTATUS};
    const CStringDictionary::TStringId FORTE_OPCUA_SERVER::scm_anDataOutputTypeIds[] = {g_nStringIdBOOL, g_nStringIdWSTRING};

    void FORTE_OPCUA_SERVER::executeEvent(int pa_nEIID){
        switch(pa_nEIID){
            case scm_nEventINITID:
                #error add code for INIT event!
                break;
            case scm_nEventREQID:
                #error add code for REQ event!
                break;
        }
    }

```

Figura 31: Estructura en C++ de SIFB OPCUA_SERVER

Utilizando la herramienta Eclipse CDT C/C++ se genera un proyecto. En este proyecto se declaran y definen métodos IEC 61499 que enlazan los algoritmos en el hardware; también incluye las funciones para enlazar código C/C++ con código FORTE, como el que se muestra en la Figura 32.

```

FORTE_OPCUA_SERVER::FORTE_OPCUA_SERVER(const CStringDictionary::TStringId pa_nInstanceNameId, CResource *pa_poSrcRes) : \
    CFunctionBlock( pa_poSrcRes, &scm_stFBInterfaceSpec, pa_nInstanceNameId, m_anFBConnData, m_anFBVarsData){
    memset(m_servercfgfile, 0, sizeof(m_servercfgfile));
    memset(m_servername, 0, sizeof(m_servername));
    m_opcuaserver = static_cast<COPCUAServer*>(NULL);
    strncpy(m_servername, CStringDictionary::getInstance().get(pa_nInstanceNameId), sizeof(m_servername));
}

FORTE_OPCUA_SERVER::~FORTE_OPCUA_SERVER(){
    if(m_opcuaserver != static_cast<COPCUAServer*>(NULL))
    {
        if(m_opcuaserver->isConnected())
        {
            m_opcuaserver->Stop();
            OPCUAMdk::GetInstance().DeleteServer(m_opcuaserver);
        }
    }
    m_opcuaserver = static_cast<COPCUAServer*>(NULL);
    memset(m_servername, 0, sizeof(m_servername));
    memset(m_servercfgfile, 0, sizeof(m_servercfgfile));
}

```

Figura 32: Algoritmo en C++ para integración en FORTE de SIFB OPCUA_SERVER

Para el caso específico de un SIFB de comunicaciones, en este punto se programa el acceso a los protocolos de comunicación deseados e incluye el código fuente del runtime para enlazar con el código de comunicaciones desarrollado. Este código se analiza y crea un fichero ejecutable, que incluye los SIFBs desarrollados, generando un nuevo

runtime. Este nuevo runtime FORTE integra SIFBs, el cual implementa y cumple con los servicios del conjunto de SIFBs.

3.6 MODELADO DE LA ARQUITECTURA CPPS BAJO LA NORMA IEC 61499

En el modelo propuesto para una arquitectura CPPS no se especifican funciones detalladas para cada modelo de la arquitectura y tampoco los tipos de los datos relacionados con esas funciones. Por lo tanto, para obtener datos de los diferentes modelos en la arquitectura CPPS es necesario utilizar protocolos de comunicación. En este trabajo se propone la utilización del estándar de comunicación ofrecido por OPC-UA. Este supone una alternativa prometedora ya que entre sus ventajas destaca una arquitectura orientada a servicios, ofreciendo seguridad de datos y modelado de información de manera robusta.

Por otra parte, el modelo IEC 61499 está diseñado para fines generales y, por lo tanto, necesita alguna modificación antes de ser implementado en aplicaciones reales.

Debido a ello, tiene sentido combinar los dos modelos para, de esta manera, implementar con redes de FBs una arquitectura CPPS flexible. La arquitectura CPPS propuesta integra componentes estándar y reutilizables, tales como servicios de comunicación basados en OPC-UA, algoritmos de control, software, etc. La arquitectura basada en redes FBs también es deseable para diseñar, desarrollar e implementar CPPSs altamente rentables en industriales reales. La Tabla 7 presenta las relaciones que permite combinar elementos de ambas arquitecturas.

Tabla 7: Relación entre los modelos de la arquitectura CPPS e IEC 61499

ARQUITECTURA CPPS	MODELO IEC 61499
<i>Physical Device</i>	Dispositivo
<i>Logical Device (LD)</i>	Recurso
<i>Logical Process Node (LPN)</i>	CFB
<i>Information Component (IC)</i>	CFB
<i>Data Object</i>	FB Básico, SIFB
<i>Atomic Services</i>	FBs, CFB

3.7 OPC-UA E IEC 61499 PARA INTEGRACIÓN DE COMUNICACIONES EN ARQUITECTURAS CPPS

Normalmente, un perfil para OPC-UA está diseñado para la integración vertical de sistemas de automatización y control industrial. El uso de una arquitectura OPC-UA permite una descripción completa de cualquier dato del sistema de automatización, independientemente de su complejidad. En este sentido, la información descrita en el Espacio de Direcciones del servidor OPC-UA puede ser suficiente para la mayoría de los sistemas industriales. A continuación, se explica la integración vertical a nivel de planta usando IEC 61499 como herramienta para implementar e integrar datos de procesos usando OPC UA como protocolo de comunicación dentro de la arquitectura CPPS.

3.7.1 Conjunto de SIFBs para OPC-UA

En esta sección se presenta el desarrollo de un conjunto de SIFBs que encapsulan las operaciones que ofrecen servicios de OPC-UA, siguiendo la metodología previamente explicada.

3.7.1.1 SIFB OPCUA_SERVER

Mediante la utilización de este SIFB es posible gestionar la configuración y el funcionamiento de un servidor OPC-UA estático. El servidor OPC-UA se configura empleando un archivo en formato XML y permite integrar los dispositivos a nivel de planta usando protocolos nativos de redes industriales como por ejemplo Modbus TCP, comunicaciones S7, etc. Este archivo sigue la arquitectura CPPS explicada anteriormente e incluye todos los parámetros esenciales para el funcionamiento y configuración del servidor OPC-UA como son: la dirección URL, el identificador URI, nombre del proveedor, el nombre del servidor, la versión, etc. El esquema del archivo de configuración XML se presenta en la Figura 33.

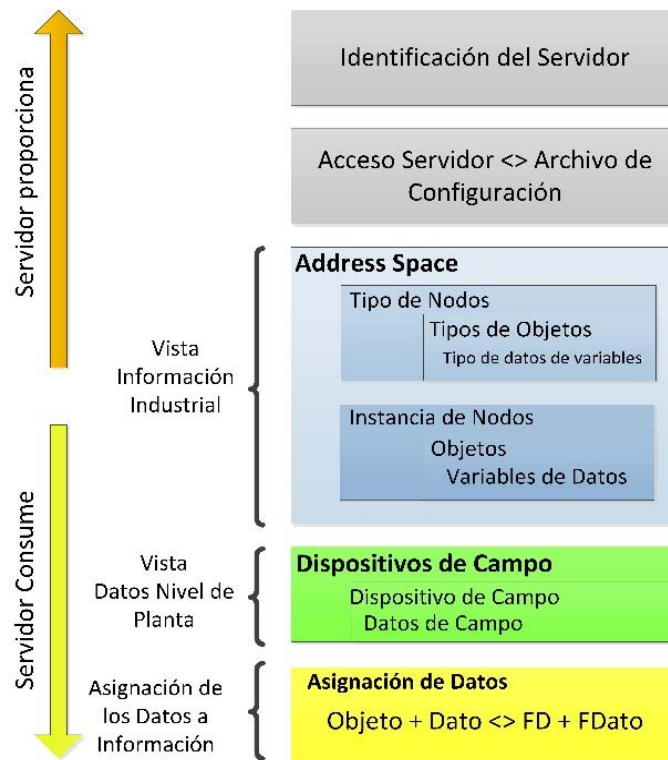


Figura 33: Modelo de implementación de servidor OPC-UA

En este archivo de configuración XML declara lo siguiente:

- a) **Server ID (Identificación, Información del servidor)**: Esta sección da información básica del servidor OPC-UA como por ejemplo el dominio y la versión del mismo.
- b) **Acceso al Servidor (Server Access)**: Este elemento proporciona el acceso al servidor OPC-UA, debido a que permite ingresar el nombre del servidor, la dirección URL, el identificador URI. El archivo de configuración depende de la pila de comunicación usada para la implementación del servidor.
- c) **Espacio de Direcciones (NodeTypes, NodeInstances)**: Esta sección declara los nodos OPC-UA (tipos e instancias) que constituyen el modelo de información correspondiente a los nodos lógicos de proceso (LPNs), así como los datos lógicos de proceso (tags) disponibles en cada uno de los nodos. A su vez esta sección se compone de dos grupos de elementos: (i) Tipos de Nodos (*NodeType*), aquí se declaran los Tipos de Objetos (*ObjectTypes*) y los Tipos

de Variables de Datos (*DataVariableTypes*) asociados a ellos. (ii) Los elementos de las Instancias de Nodos (*NodeInstance*) están incluidos en las Instancias de Objetos (*ObjectInstances*) – estas son instancias que ya han sido declaradas en los Tipos de Objetos (*ObjectTypes*) – y las Variables de Datos (*DataVariables*) los cuales son instancias de los Tipos de Variables (*DataVariableTypes*). Las Instancias de Objetos (*ObjectInstances*) constituyen los LPNs, mientras que las Instancias de Variables (*DataVariables*) constituyen los tags asociados a los LPNs. Lo explicado se puede ver en la Figura 34.

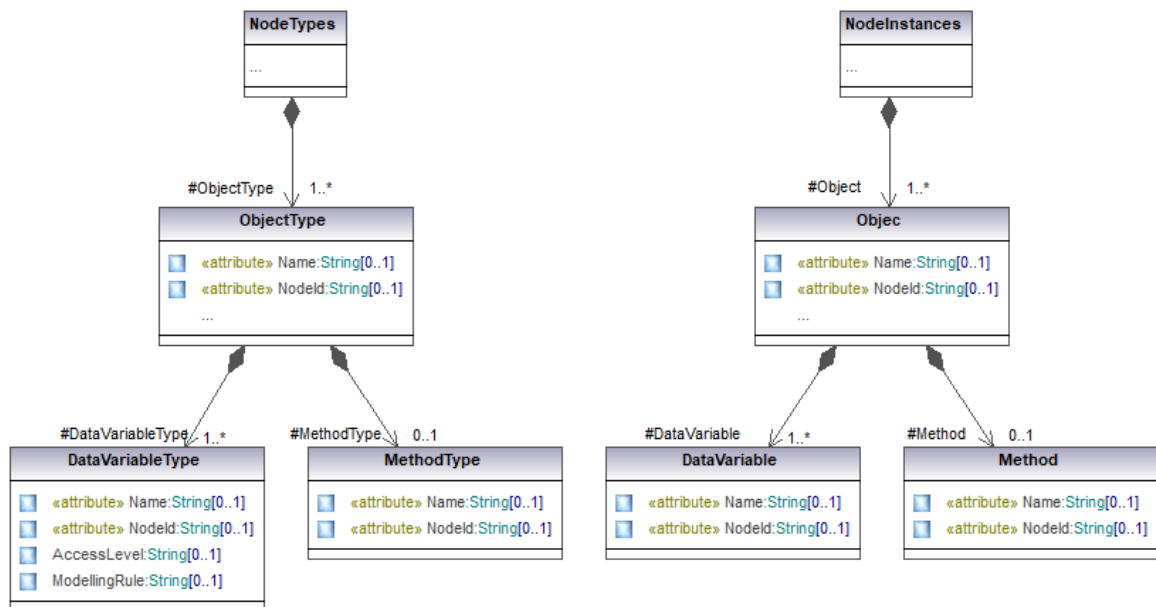


Figura 34: Meta-Modelo del Espacio de Direcciones

- d) **Dispositivos de campo (FieldDevices)**: En esta sección se agrupan las definiciones de los dispositivos de campo a los que tiene acceso el servidor. También se definen los datos de proceso (*FieldData*) suministrados por cada dispositivo. Los dispositivos de campo se caracterizan por el protocolo de comunicación o el mecanismo de acceso a los datos de proceso, y la información relacionada con el mismo. Ver Figura 35.

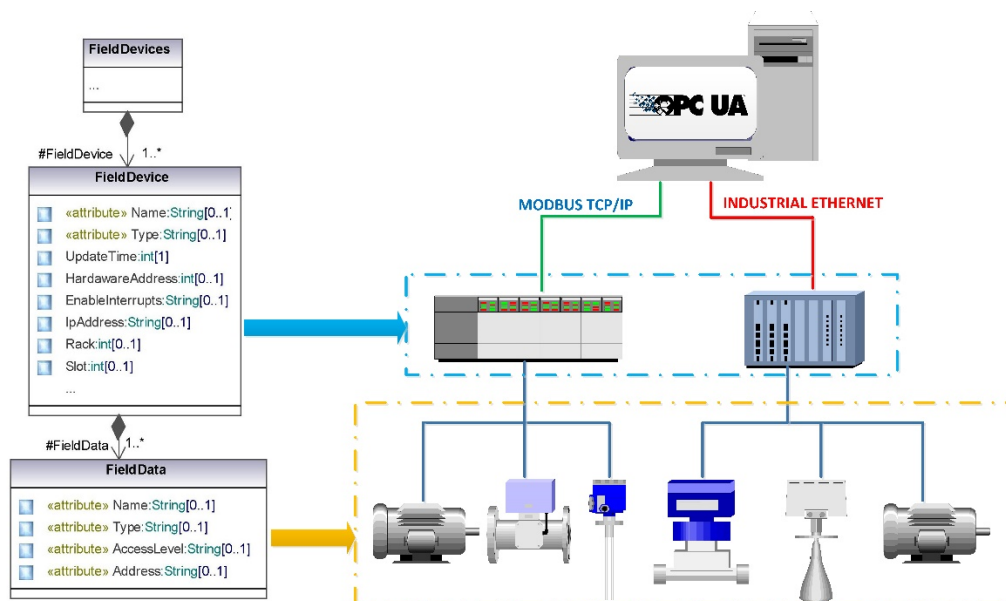


Figura 35: Modelo de Dispositivos de Campo

e) **Mapeo de Datos (DataMapping):** En esta sección se definen las relaciones existentes entre las variables de datos (*DataVariables*) ya declaradas la sección del Espacio de Direcciones y sus correspondientes datos de proceso (*FieldData*) de los dispositivos de campo (*FieldDevice*). Un ejemplo del archivo de configuración para el servidor OPC-UA en formato XML podemos verlo en la Figura 36.

```

<OPCUAServerConfig Comment="OPCUAmdk Test Server Configuration - Test 04">
  <Identification ApplicationDomain="ThinkingFactory" Description="OPCUA server configuration model" />
  <NodeTypes>...</NodeTypes>
  <NodeInstances>...</NodeInstances>
  <FieldDevices>
    <FieldDevice Name="WPA01 " Type="ModbusTCP">
      <UpdateTime>200</UpdateTime>
      <IPAddress>192.168.2.100</IPAddress>
      <Rack>3</Rack>
      <Slot>2</Slot>
      <FieldData Name="TIT-WPA01-01" Type="DOUBLE" AccessLevel="MANDATORY" Address="%IB0" />
      <FieldData Name="TIT-WPA01-02" Type="DOUBLE" AccessLevel="MANDATORY" Address="%IB1" />
      <FieldData Name="PIT-WPA01-01" Type="DOUBLE" AccessLevel="MANDATORY" Address="%IB2" />
      <FieldData Name="PT-BESPA1-01" Type="DOUBLE" AccessLevel="MANDATORY" Address="%IB3" />
      <FieldData Name="PT-BESPA1-02" Type="DOUBLE" AccessLevel="MANDATORY" Address="%IB4" />
      <FieldData Name="TT-BESPA1-01" Type="DOUBLE" AccessLevel="MANDATORY" Address="%IB5" />
      <FieldData Name="TT-BESPA1-03" Type="DOUBLE" AccessLevel="MANDATORY" Address="%IB6" />
      <FieldData Name="VT-BESPA1-01" Type="DOUBLE" AccessLevel="MANDATORY" Address="%IB7" />
    </FieldDevice>
  </FieldDevices>
  <DataMappings>...</DataMappings>
</OPCUAServerConfig>
  
```

Figura 36: Ejemplo de Archivo de Configuración de servidor OPC-UA en formato XML

Una vez explicado el archivo de configuración del servidor OPC-UA, se procede a explicar el SIFB que permite realizar la comunicación a nivel de campo integrándolos con el protocolo OPC-UA. Además de eventos comunes de SIFBs en la norma IEC 61499 como INIT, REQ, INITO y CNF explicados anteriormente (Figura 37), el SIFB OPCUA_SERVER también presenta los siguientes parámetros de entrada y salida:

- QI (BOOL): Este dato de entrada trabaja conjuntamente con el evento INIT para conectar o desconectar el servidor OPC-UA. Si se solicita el evento INIT, y si QI es VERDADERO (TRUE), el servidor OPC-UA intenta conectar; si QI es FALSO (FALSE), el servidor OPC-UA finaliza su ejecución.
- CONFIGFILE (WSTRING): Contienen el nombre completo del archivo XML de configuración.
- QO (BOOL): Informa sobre el resultado del último proceso ejecutado.
- STATUS (STRING): Ofrece información del estado del servidor.

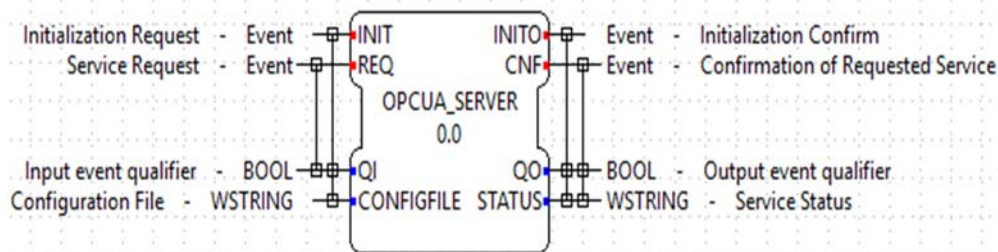


Figura 37: SIFB OPCUA_SERVER.

3.7.1.2 SIFB SERVER OPC-UA Dinámico

Este SIFB permite realizar la comunicación OPC-UA usando la capa nativa de comunicación de IEC 61499 denominada FORTE CommLayer. Esta interface de comunicación de red ha sido diseñada para ser lo más flexible posible. Los pasos básicos para diseñar esta interface parten de implementar el protocolo OPC-UA con librerías desarrolladas en C++ que permitan interactuar con el diseño estándar dado por la norma. La pila de comunicación usada permite integrar OPC-UA en FORTE (4DIAC RTE), la cual es totalmente escalable, admite arquitecturas de múltiples hilos, donde cada conexión o sesión se operan por hilos separados.

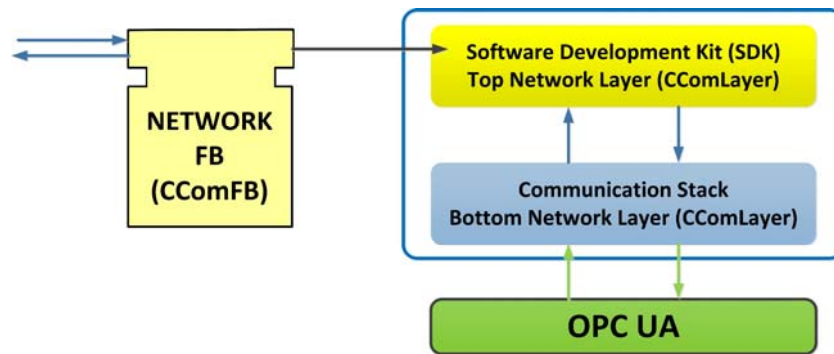


Figura 38: Arquitectura implementación SIFB de servidor OPC UA

Esta pila de comunicación se puede ejecutar en dispositivos diversos ya sea en computadores industriales o en dispositivos empotrados de bajo costo como son las tarjetas Raspberry Pi o BeagleBone Black. La arquitectura de esta implementación se muestra en la Figura 38. Aquí se puede observar los diferentes tipos de clases de FORTE y la interacción básica entre sí. La clase del Bloque de Función llamada *CCommFB* es la única clase que interactúa con las aplicaciones IEC 61499 y permite enviar y recibir eventos basados en la norma IEC 61499.

El bloque de función y la capa de comunicación implementada se relacionan usando los siguientes entradas y salidas de datos implementadas en el SIFB (Figura 39):

- ID (WSTRING): Es una entrada del SIFB que sirve para implementar todos los parámetros que necesita el protocolo OPC UA para su correcta configuración. Los parámetros se ingresan como una cadena de caracteres, existen parámetros obligatorios [*m*] y opcionales [*o*]. A continuación, se describen estos parámetros: Nombre del Servidor (*Server name*) [*o*], su modo de ingreso es *svr:<servername>* el nombre permite identificar al servicio OPC UA implementado; Dirección del archivo de configuración (*Configuration file*) [*m*] se ingresa este parámetro usando la línea de código *cnf:<configfile>* lee un archivo de memoria en formato xml el cual contiene la dirección URL, URI, autor del servidor OPC UA, etc; Nombre del Dispositivo de Campo (*Field device*) [*o*] se usa el código *fd:<fdname,driver,params,...>* ingresa el nombre del dispositivo de campo, el dri-

ver del protocolo de comunicación industrial que usamos para integrar las variables del proceso y los parámetros adicionales para su configuración; Variable del Proceso (*Process Tag*) [o] el formato de ingreso es *tg[num]:<TagName, Type, AccessLevel, FDName...>* con esta línea de código ingresamos variable del proceso seguida de un número e incluimos su nombre, el tipo de variable, su nivel de accesos, etc.

- **SD** (ANY): Permite escribir el valor en el espacio de direcciones del servidor OPC-UA. Es un parámetro de memoria de entrada (*parameter - memory tag*) [o] se ingresa usando el siguiente formato *sd[num]:<TagName, Type, AccessLevel>* al igual que en el caso anterior se ingresa la variable pero que solo se encuentra en la memoria del dispositivo empotrado que se desea crear, esta variable debe estar acompañada con un número e incluyendo su nombre, tipo y nivel de acceso.
- **RD** (ANY): Dato que proporciona el servidor al cual éste está conectado. Al igual que en la entrada anterior este parámetro es de memoria (*parameter memory tag*) [o], la línea de código de ingreso es *rd[num]:<TagName, Type, AccessLevel>*.

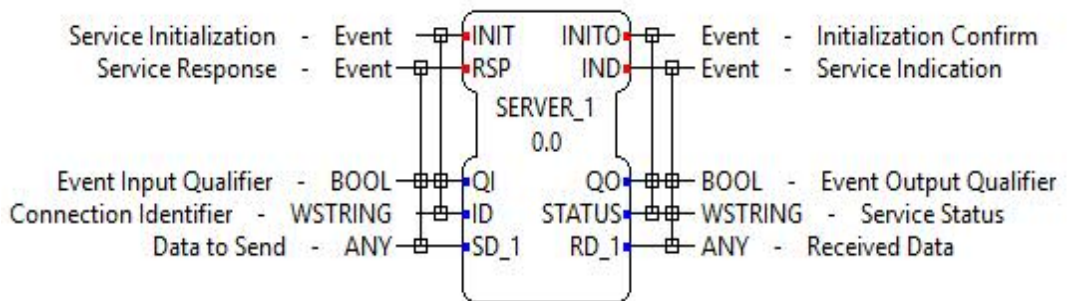


Figura 39: SIFB SERVER OPC-UA dinámico

3.7.1.3 SIFB OPCUA_CLIENT_READ

Este SIFB permite implementar un cliente OPC-UA capaz de consultar de manera síncrona variables de un servidor OPC UA. Además de los eventos comunes, este SIFB presenta los siguientes parámetros de entrada y salida (Figura 40):

- **URLSERVER** (WSTRING): Indica la URL del servidor OPC-UA.

- DATANAME (WSTRING): Nombre de la variable a acceder en el servidor OPC-UA.
- RD (ANY): Valor del dato que devuelve el servidor OPC-UA al cual el cliente está conectado. El tipo de este parámetro es ANY para mejorar su reutilización en cualquier tipo de aplicación.
- SOURCETIMESTAMP (DATE AND TIME): Indica la información de tiempo asociada al elemento en el recurso de origen.
- SERVERTIMESTAMP (DATE AND TIME): Indica la información de tiempo asociada al elemento en el servidor OPC-UA.

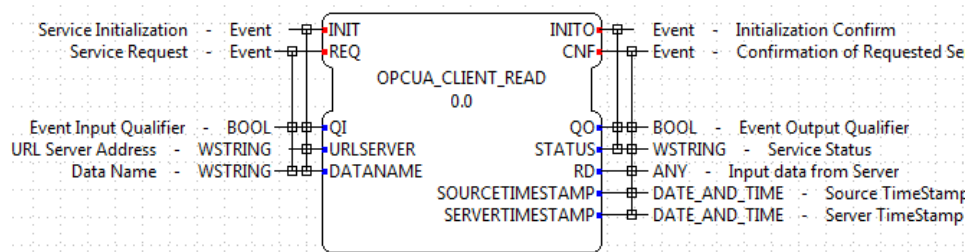


Figura 40: SIFB OPCUA_CLIENT_READ

3.7.1.4 SIFB OPCUA_CLIENT_WRITE

Mediante este SIFB (Figura 41) se crea un cliente para la escritura síncrona de variables en su servidor OPC-UA. Presenta los siguientes parámetros:

- TYPE (WSTRING): Expresa el tipo del dato para el valor que se va a escribir en el servidor OPC-UA.
- SD (ANY): Indica el valor del dato a escribir. Como en el SIFB anterior, el tipo de este parámetro es ANY con el fin de aumentar la generalidad.

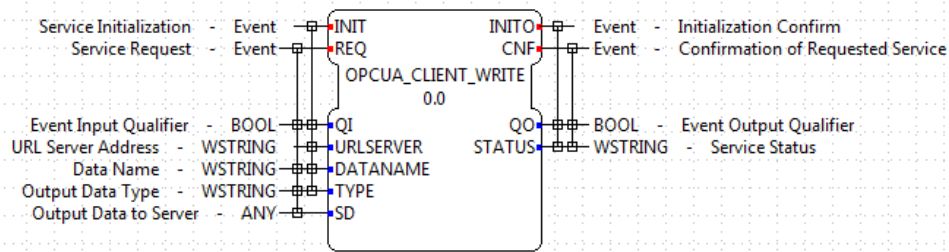


Figura 41: SIFB OPCUA_CLIENT_WRITE

3.7.1.5 SIFB OPCUA_CLIENT_SUBSCRIBE

Además de los servicios de lectura y escritura síncrona, OPC-UA permite el mecanismo de suscripción. La suscripción mantiene una copia local de los parámetros del elemento a ser supervisado. Estas copias locales pueden ser alteradas mediante la actualización de sus propiedades sin afectar el estado en el servidor. Para implementar este mecanismo se ha creado un SIFB (Figura 42) que permite crear clientes que realizan suscripción para monitorizar variables de servidores OPC-UA. Los parámetros de entrada y salida de este SIFB son:

- MODE (WSTRING): Se permiten dos modos para monitorizar las variables del servidor OPC-UA: “REPORTING” y “SAMPLING”.
- PERIOD (ULINT): Fija el período de muestreo en milisegundos para los elementos en modo de muestreo.
- RD (ANY): Proporciona el valor de los elementos monitorizados suministrados por el servidor OPC-UA.
- SOURCETIMESTAMP (DATE AND TIME): Indica el Timestamp asociado al elemento del recurso de origen.
- SERVERTIMESTAMP (DATE AND TIME): Muestra el Timestamp asociado al elemento del servidor OPC-UA.

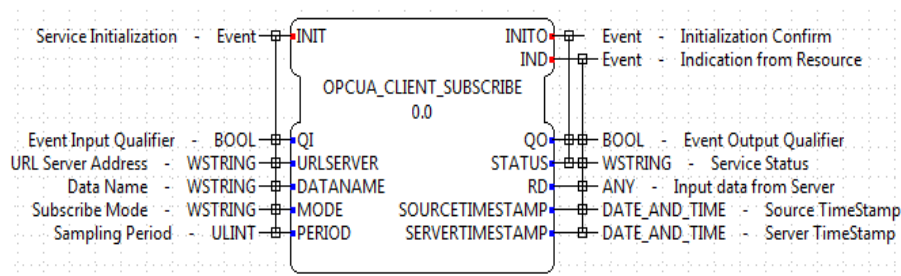


Figura 42: SIFB OPCUA_CLIENT_SUBSCRIBE

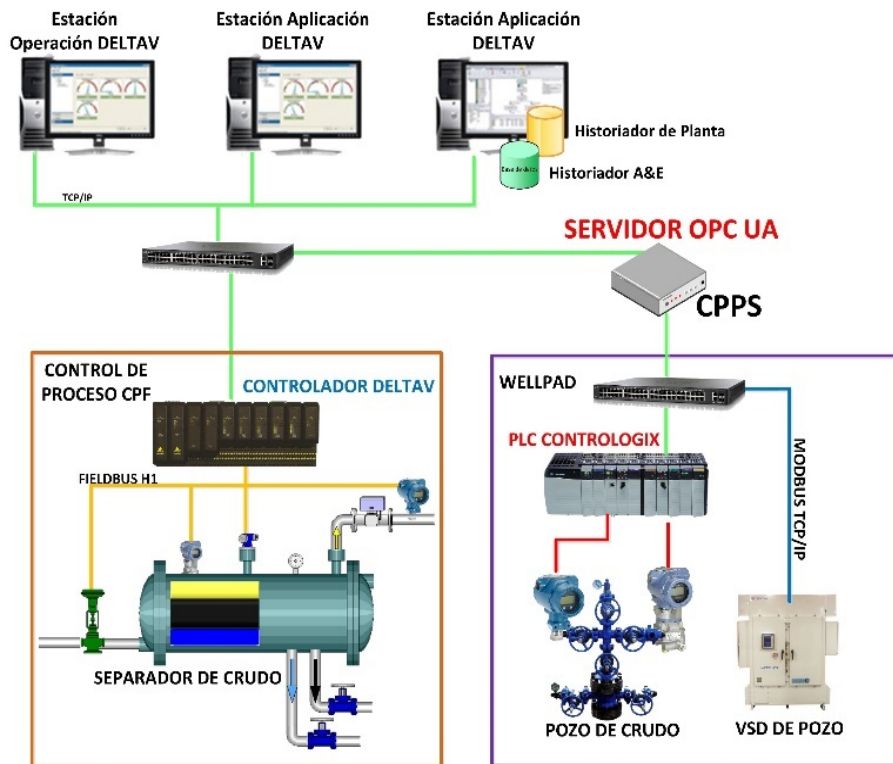


Figura 44: Sistema de comunicación simulado para el proceso de separación y extracción de crudo

la temperatura y el PIT-WPAA01-01 la presión del crudo extraído. Asimismo, cuenta con una válvula de emergencia de producción, la SDV-WPAA01-01.

Debido a que las presiones naturales de los yacimientos en Ecuador no brindan una presión suficiente para que el crudo llegue a superficie, cada pozo utiliza un sistema de bombeo electro-sumergible BESPA1, que se corresponde con el elemento M de la Figura 43, el equipo BES posee dos transmisores de presión: PT-BESPA1-01 para la presión de entrada de la bomba (*intake*) y el PT-BESPA1-02 para la presión de descarga de la bomba (*exhaust*); tres transmisores de temperatura: TT-BESPA1-01 para la temperatura del fluido, el TT-BESPA1-02 para la del motor y el TT-BESPA1-03 para la del fluido de salida; un sensor que mide la corriente del motor, CT-BESPA1-01; la tensión del motor se mide mediante VT-BESPA1-01 y el FT-BESPA1-01 indica el valor de la frecuencia del variador de frecuencia, VSD.

Además, el núcleo del sistema ESP está compuesto por un motor (M-WPA01-1), bomba multi-etapa (P-WPA01) y un variador de velocidad VSD (VSD-WPA01). La Figura 45 muestra el sistema de comunicación actual en el que se encuentra el servidor OPC-UA.

El diseño y desarrollo de un ICs es un proceso estructurado, pero simple, si se siguen los siguientes pasos:

Paso 1) Descomponer el proceso de adquisición de las variables de campo de una manera detallada y jerárquica para poder ser representada con los FBs desarrollados para OPC-UA y los existentes de la norma IEC 61499. *Paso 2)* Seleccionar los FBs correspondientes de la biblioteca de OPC-UA o los FB estándares para todas las sub-funciones

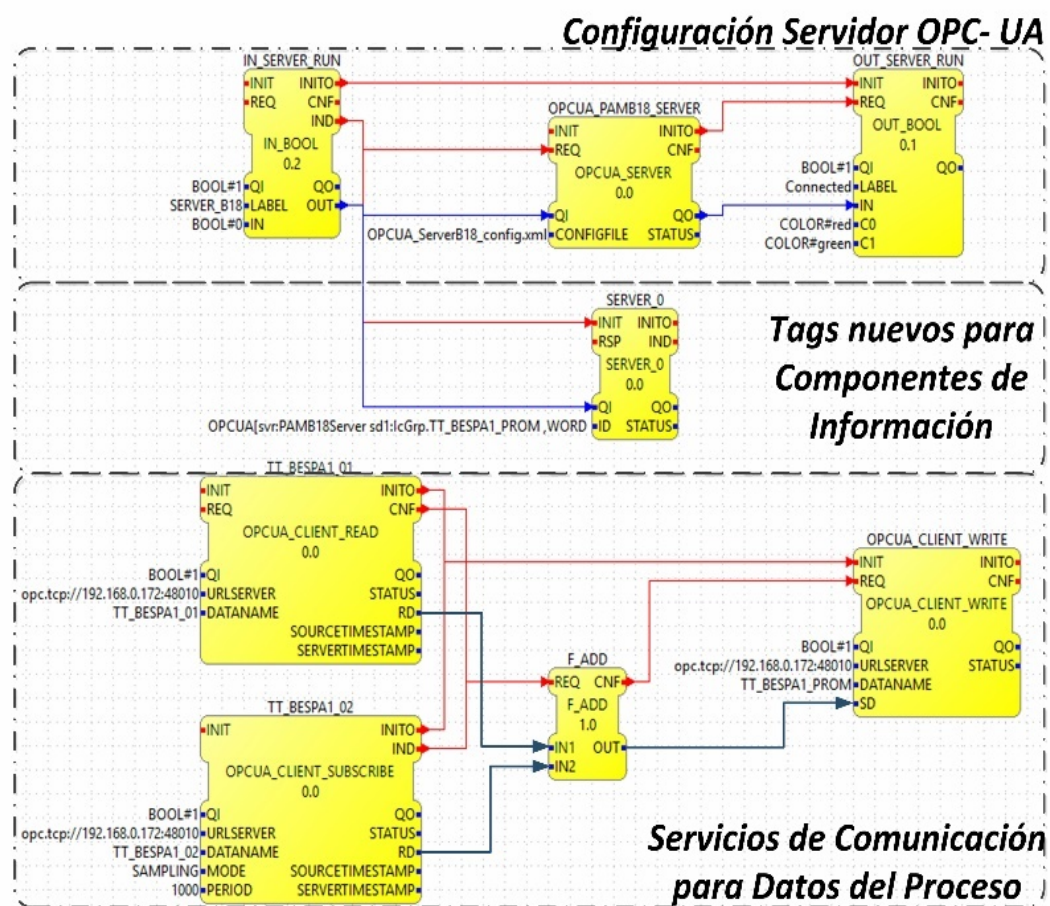


Figura 45: Componente de Información (ICs) de Pozo PAA001

diseñadas en el paso anterior, luego se deben crear las instancias de los FB seleccionados, configurar sus parámetros y conectar estos FB con flujo de datos y flujo de eventos. Paso 3) Validar todo el modelo de ICs para evitar errores. Paso 4) Integrar los componentes físicos a los ICs y realizar pruebas completas de comunicación de los diferentes tags.

Como ejemplo de caso de estudio, se muestra el diseño de un IC usando los FBs estándar y FBs para comunicación OPC-UA desarrollado en este trabajo. La Figura 45 muestra la arquitectura e una red de FBs del IC. El IC consta de tres módulos principales: i) el servidor estático de OPC-UA, el cual, lee la configuración del mismo en formato XML; ii) la creación dinámica de nuevas etiquetas a variables del proceso utilizando el FB servidor que utiliza la capa de comunicación de FORTE y, si se requiere, utilizar FBs de Servicios Atómicos (Almacenamiento de Datos, Reporte de Datos, Lectura de Datos, etc.) y iii) finalmente, una vez creados los ICs, se debe realizar integración de servicios de comunicación con los dispositivos de campo utilizando los FBs clientes (Escribir, Leer, Suscribirse) de OPC-UA.

3.9 CONCLUSIONES

Este capítulo presenta una aproximación al acceso a datos de campo en sistemas de control de procesos utilizando servidores OPC-UA en arquitecturas CPPS de bajo coste utilizando aplicaciones bajo la norma IEC 61499. En este caso, la aplicación de esta arquitectura se ha dirigido hacia la industria de petróleo y gas.

El uso de este tipo de arquitecturas ayuda a introducir elementos novedosos sobre CPPS dentro del paradigma de la Industria 4.0. La arquitectura propuesta proporciona una infraestructura M2M para las comunicaciones a nivel de planta y la integración en el proceso productivo de dispositivos de niveles superiores.

Se propone el uso de un conjunto de SIFBs para implementar servidores y clientes OPC-UA, incluyendo mecanismos de suscripción. Estos bloques de función permiten, de una manera rápida y sencilla, la construcción de nuevas aplicaciones distribuidas basadas en componentes mediante entornos de programación bajo la norma IEC 61499.

La utilización de bloques de función para diseñar componentes de información proporciona beneficios evidentes y potenciales. Quizás el más simple, pero más importante, es que el mismo componente de información, el cual, además de ser reutilizable, es aplicable en diferentes procesos industriales mediante la reconfiguración del software. También es factible añadir, eliminar y reemplazar algunas funciones de un componente de información en producción cuando lo soliciten los requisitos de la aplicación.

4

MODELADO DE APLICACIONES DISTRIBUIDAS BAJO ISA 88/95

Una de las bases sobre las que se asienta la cuarta revolución industrial es la accesibilidad a cualquier información relevante en tiempo real. Esto se alcanza conectando todas las instancias que conforman la cadena de valor. Lo que se persigue, en general, es la gestión de todo el proceso de la cadena de valor, que derivará en una producción más eficiente dando lugar a mejores productos y servicios. Las redes verticales tienen su fundamento en la utilización de Sistemas de Producción Ciberfísicos (CPPSs) y en las organizaciones que posibilitan la integración vertical que va desde los sensores hasta el nivel de empresa de la compañía.

Sin embargo, la interoperabilidad global solo se garantiza si se utilizan estándares industriales para modelar las vistas de las partes implicadas y comunicando los dispositivos presentes, diversos y heterogéneos.

Para ello en este capítulo se presenta una propuesta de metodología de modelado basada en estándares industriales como son las normas ISA 88/95. El objetivo es generar la configuración estática del servidor OPC-UA a partir de los modelos de la planta y de los dispositivos que suministran los datos.

A continuación, se expone uno de los apartados más relevantes de este trabajo, que presenta la principal aportación, una herramienta software para diseño de aplicaciones de control distribuidas teniendo como base el estándar IEC 61499. El compositor captura la información necesaria para crear la interconexión entre los componentes distribuidos de la aplicación y generar el modelo del sistema completo. Por tanto, se encarga de incluir de forma automática los SIFBs para protocolo OPC-UA desarrollados en el capítulo anterior, las conexiones y los datos de configuración de la aplicación.

4.1 TRABAJOS RELACIONADOS

Considerando las necesidades identificadas en el sector de producción del petróleo, este capítulo presenta el trabajo que se está realizando en relación con el reto que supone la monitorización remota de este proceso particular. Se presenta aquí una propuesta genérica que facilita el acceso remoto a los datos del proceso, alcanzando una integración vertical desde las áreas de empresa o negocio hasta el nivel de producción en planta. Para este propósito y siguiendo la tendencia recomendada por Industria 4.0, se han seleccionado los estándares, OPC-UA (IEC 62541) [56] e ISA 95 (IEC 62264) [128] e ISA 88 (IEC 61512) [129] como columna vertebral para la comunicación y modelado de planta respectivamente.

En esta línea, pueden destacarse proyectos de investigación que utilizan el estándar ISA 95 para la definición de arquitecturas. Es el caso de HoVer [13], que aborda la integración de los flujos de información horizontal y vertical, donde los escasos aspectos de la integración vertical que se abordan lo hacen utilizando conceptos de ISA 95, aunque únicamente como extensión del núcleo, centrado en la horizontal. Otro ejemplo es [130]

que aborda la monitorización desde un entorno orientado a servicios (SOA) y con organización basada en Redes de Petri de alto nivel. De ISA 95, en este caso, solo se utiliza el modelo que identifica los niveles de la empresa.

Dimitrova [131] propone una arquitectura abierta basada en ISA 88 para fabricación flexible. Para ello, diseñan una aproximación modular que minimiza los esfuerzos de programación en el sistema de automatización que exige un nuevo producto. Además, las interfaces entre los módulos se basan en OPC-UA. Sin embargo, la integración vertical no está entre sus objetivos.

Varios investigadores comparten sus ideas sobre el desarrollo y aplicabilidad de la norma IEC 61499 para mejorar el control dentro de los procesos industriales. En [58] y [132] los autores presentan un enfoque sobre los conceptos básicos de la norma IEC 61499 y su uso en paralelo con el estándar ISA 88 para desarrollar sistemas de control eficientes para procesos industriales. La propuesta de control de este artículo pretende establecer una forma de migrar de la norma IEC 61131 a un modelo flexible a través de una aproximación formada por tres partes: i) el desarrollo de componentes funcionales, ii) la descripción de un modelo lógico a través de modelos SIPN (Signal Interpreted Petri Nets) y iii) el mapeo del modelo SPIN a una red de FBs. Otra propuesta similar presentada en [133] muestra una aproximación que integra tres modelos diferentes: IEC 61499, ISA 88 y SIPN; para obtener un control reconfigurable basado en componentes de software reutilizables.

Por otra parte, en [134] los autores definen 4 retos que debe resolver el uso conjunto de la norma IEC 61499 e ISA 88/95 para la aceptación a nivel industrial, estos retos son: escalabilidad, mantenibilidad, extensibilidad y predictibilidad. A través de los años, estos desafíos han sido continuamente superados gracias a la contribución de varios grupos de investigación que han estado desarrollando herramientas de ingeniería para solucionar estos inconvenientes. Un ejemplo específico en el cual se usa ISA 88 e IEC 61499 es el caso de estudio presentado en [135] donde los autores no sólo analizan la factibilidad de estas normas en cuanto al punto de vista de la ingeniería de automatización, sino que también discuten sobre los desafíos y las ventajas de su aceptación al decidir

implementarla en industrias tales como la industria aeroespacial. Otro ejemplo, que muestra nuevas aplicaciones para el uso de la norma IEC 61499 es la investigación realizada por [136] donde los autores mejoran el nivel de paralelismo al ejecutar los FBs con el fin de mejorar el rendimiento del sistema industrial y reducir el tiempo de respuesta a los eventos de entrada.

Se han realizado varios trabajos académicos para favorecer el intercambio de información estandarizada según los modelos del ISA 88/95, así como diferentes formas de implementación. Esto ha permitido que a nivel industrial exista una mejora continua en varias áreas tales como: ahorro de costes [137], optimización del tiempo de producción [6], [138], [139], aumento en la reutilización de código en dispositivos de control para procesos de fabricación por lotes [140], mayor coherencia en la ejecución y calidad del proceso [141], crecimiento de la capacidad de producción, así como una reducción de la cantidad de operadores. [142]

Sin embargo, si bien muchos estudios se enfocan en diseñar nuevos sistemas y herramientas basados en las normas ISA 88/95 [137], pocos investigadores intentan analizar los sistemas ya existentes y proveer un informe sobre su adecuación al estándar, o estudiar qué tipo de información de fabricación contienen los sistemas, a la luz de la clasificación propuesta en ISA 88/95. La realización de esto favorecería la integración entre sistemas sin obligar a las empresas a cambiar radicalmente su forma de trabajar, como así también proveer un marco para el análisis de los sistemas existentes y posibles formas de modificarlos, con el objetivo de adecuarse a las normas ISA 88/95.

En trabajos como [143] los autores presentaron una arquitectura general para la monitorización remota. Ésta consiste en una propuesta basada en modelos donde se identifican los diferentes modelos necesarios para alcanzar la integración vertical. En este capítulo se va más allá, pues aplica dicha arquitectura genérica al caso particular de la industria de producción del petróleo. Para ello se definen unos meta-modelos específicos basados en estándares industriales (ISA95 e ISA88) para el proceso de producción y para los dispositivos inteligentes encargados de suministrar la información.

4.2 ESTÁNDARES INDUSTRIALES

4.2.1 Norma ISA 95

En la actualidad los clientes son más exigentes en lo que se refiere a precios, servicio y calidad; esto obliga a las empresas a conocer mejor las necesidades de los usuarios, bajar los costes, reducir tiempos de producción y mejorar la calidad de sus productos. Para enfrentar estos desafíos, en ocasiones no basta con contar con una amplia variedad de personal bien entrenado; las compañías se han visto obligadas a convertirse en empresas ágiles y eficientes usando nuevas técnicas y procesos para monitorización de datos, así como, realizando mejoras en los procesos internos. Lo cual implica optar por el uso de técnicas y tecnologías que sean flexibles, de buen rendimiento y que posean herramientas que permitan una adopción rápida por parte de sus trabajadores, desarrolladores y administradores. [144]

ISA 95 es la norma internacional para la integración de los sistemas administrativos de las empresas y sistemas de control y automatización industrial. Fue desarrollada por ISA (International Society of Automation) en el año de 1990, con el fin de reducir el riesgo, el coste y los errores que van de la mano con la implementación de interfaces entre sistemas empresariales y sistemas de control. Actualmente está aceptado como norma IEC 62264.

ISA 95 contiene modelos y terminología que pueden ser usadas para determinar qué información se debe intercambiar entre las diferentes funciones empresariales (compras, ventas, finanzas, logística, mercadeo) y las operaciones de administración de fabricación (producción, inventarios, mantenimiento y calidad) a través de niveles de jerarquías que permiten la integración de la industria con el sistema de control. Esta información se estructura en modelos UML, que son la base para el desarrollo de interfaces estándar entre los sistemas ERP y MES. El estándar ISA 95 se puede utilizar para varios fines, por ejemplo: Como una guía para la definición de los requisitos del usuario, para la selección de proveedores de MES y como una base para el desarrollo de sistemas MES y bases de datos.

ISA 95 proporciona una organización jerárquica del sistema de producción, que va desde los procesos físicos, bien sean por lotes, continuos o discretos, hasta el nivel más alto llamado de Logística de Negocio [145]. La mayor parte de la información descrita en esta norma se puede agrupar en tres grandes áreas:

- Información necesaria para producir un producto.
- Información sobre la capacidad para producir un producto.
- Información sobre la producción real de un producto.

El propósito del estándar es acentuar las buenas prácticas en la integración de sistemas de control con sistemas empresariales durante todo su ciclo de vida, se puede utilizar para la mejora de las capacidades de integraciones existente y se puede aplicar sin importar el grado de automatización de los sistemas. El estándar ISA 95 está dividido en 5 partes (ver Figura 46):

- La parte 1 provee la terminología estándar y la definición de modelos de objetos, que pueden ser utilizados para decidir qué información debe ser intercambiada.

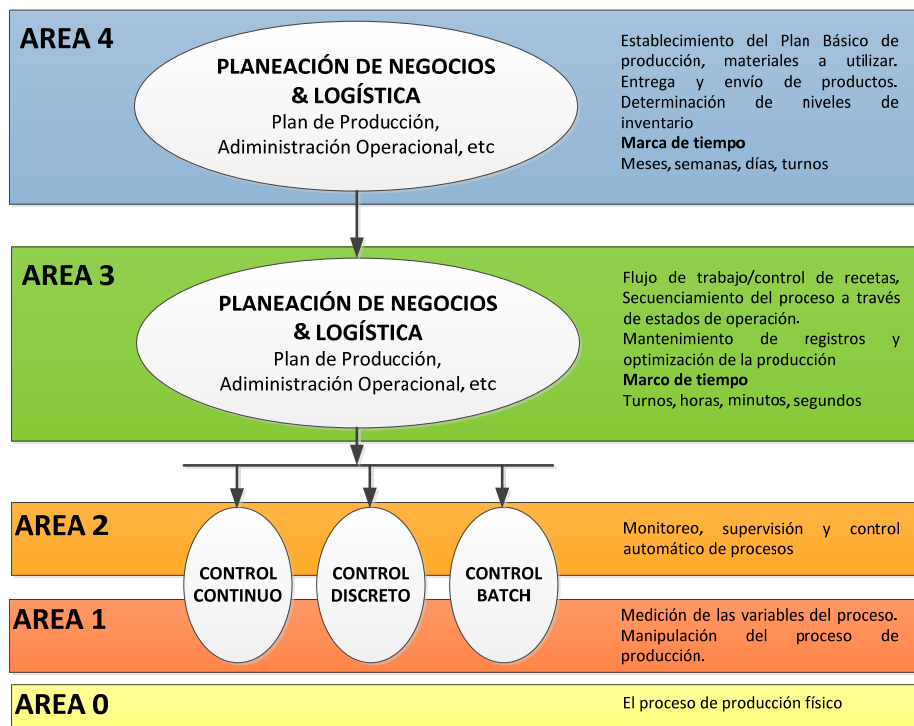


Figura 46: Modelo intercambio de información

- La parte 2 consta de los atributos para cada objeto que se define en la parte 1. Los objetos y los atributos de la parte 2 se pueden utilizar para el intercambio de información entre diferentes sistemas y sistemas diferentes; pero estos objetos y atributos también se pueden utilizar como base para bases de datos relacionales.
- La parte 3 se centra en las funciones y actividades en el nivel 3 o capa Producción/MES (calidad, mantenimiento, inventarios y producción).
- Las partes 4 y 5 proporcionan, como sus títulos lo indican, los “Modelos de objetos y atributos de la gestión de las operaciones de manufactura” y la “Integración de la empresa y los sistemas de control”, respectivamente.

Por los beneficios ofrecidos, son varias las empresas a nivel mundial que se han adaptado a este estándar, obteniendo resultados muy satisfactorios a la hora de organizar su información. Por otra parte, en el ámbito académico, principalmente en las universidades cuyos programas tratan temas relacionados con la automatización e integración empresarial, han adoptado la Norma ISA-95 como elemento de estudio, teniendo como único soporte las bases teóricas que aporta este estándar.

4.2.2 Norma ISA 88

La norma ISA 88 desarrolló una metodología aplicable a empresas que posea procesos de fabricación secuenciales (batch). Permite a la industria diseñar modelos para automatizar la producción y al mismo tiempo, reducir tanto la complejidad como los costos asociados a los sistemas propietarios, aun cuando, al inicio era pensada solo para la industria por lotes actualmente se puede usar para todo cualquier tipo de industrias.

La norma ISA 88 indica que se debe analizar por separado, las capacidades de los equipos de la planta (modelo físico) y los procedimientos de control requeridos, llamados recetas, para realizar un proceso de control industrial. El estándar pretende también aprovechar al máximo, la capacidad instalada y otorgar flexibilidad al momento de editar los algoritmos de control.

ISA 88 ha sido la base principal para desarrollar herramientas software encargadas de automatizar procesos de producción [146] y se ha reportado una variedad de investigaciones usando la norma para mejorar la integración de procesos de producción [132], [147], y [148]. Sin embargo, cuando se utiliza los modelos proporcionados por esta norma para diseñar herramientas software, se observa que en el estándar se describen pautas y definiciones para cada uno de los componentes que hacen parte de los modelos, pero así mismo, dicho estándar no es claro con respecto de lo que abarca cada modelo dentro del proceso, generando varios problemas que no han sido reportadas en la literatura ni mucho menos atendidos por investigaciones académicas.

ISA 88 define los siguientes modelos: modelo físico, modelo de control de procedimientos y el modelo de recetas. La norma a tratado de abordar los siguientes problemas: La falta de un modelo universal para el control de lotes, la dificultad en la comunicación de las necesidades del usuario, la integración entre proveedores de automatización de proceso por lotes, y la dificultad en la configuración de control de lotes. [129]

El principal objetivo del modelo físico propuesto por el estándar ISA-88, el cual será usado en el desarrollo del presente capítulo, es estructurar los activos físicos de la empresa, incluyendo equipos, talento humano y el financiero. Esta subdivisión debe realizarse teniendo en cuenta las características y perfiles de los recursos que se necesitan para desarrollar cada una de las operaciones y acciones del proceso de producción, debiéndose identificar claramente la capacidad de procesamiento, almacenamiento y la función específica que cumplen dentro de la cadena de valor. El modelo físico se utilizará para la integración vertical con servidores OPC-UA.

4.3 PROPUESTA DE META-MODELO DE TOPOLOGÍA DE PLANTA (PTM) BASADOS EN ISA 88/95

Uno de los aspectos más críticos en un proyecto de automatización es la jerarquización del proceso y el grado de independencia que se logre entre sus actividades de procesamiento, lo cual convierte a la división física del sistema en una actividad compleja y altamente dependiente de los requisitos del medio específico donde se realiza el proceso

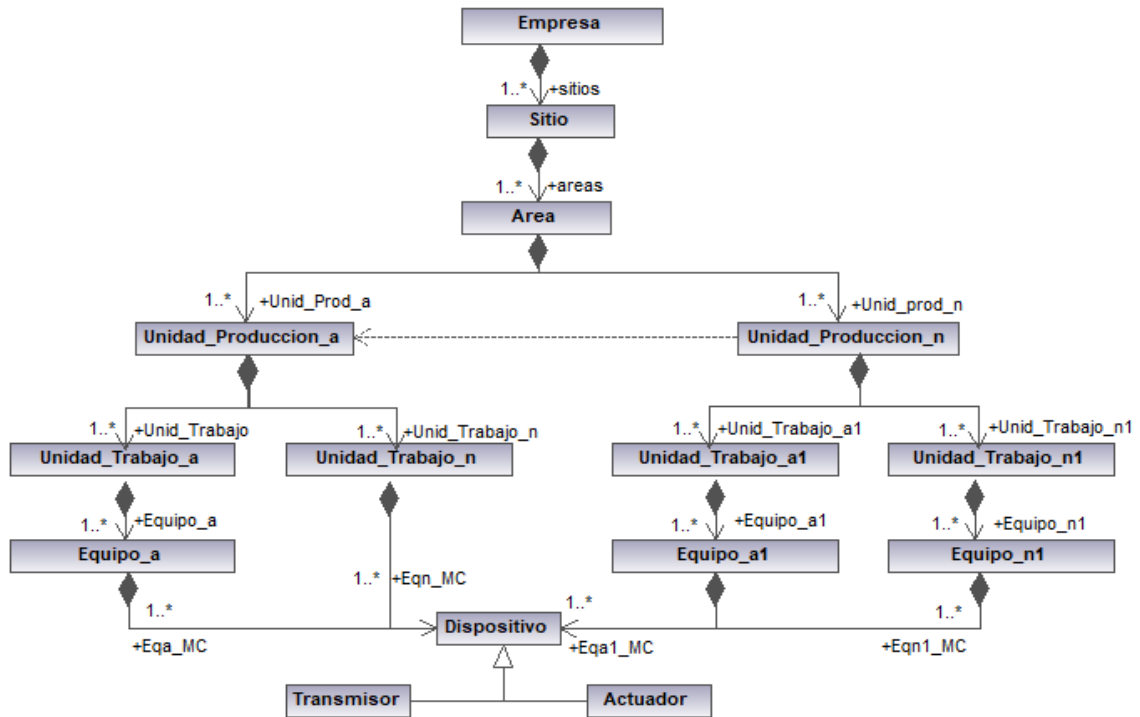


Figura 47: Sintaxis del Meta-Modelo de Topología de Planta

de producción. Es así como, una subdivisión inconsistente o inapropiada puede comprometer la efectividad de la implementación modular de los procesos de producción detallados en las recetas. [137]

El estándar ISA 95 y el modelo físico de ISA 88 proponen un modelo jerárquico donde se caracteriza el rol que toma el equipamiento en los procesos de producción. Partiendo de estos modelos generales, válidos tanto para procesos por lotes, continuos como discretos, se presentan a continuación unos modelos adaptados al caso de proceso continuo, propio del sector de producción del petróleo. A continuación, se explicará cada uno de los niveles que corresponde al Meta-Modelo que se propone para PTM (ver Figura 47):

- **Empresa (Enterprise).** La empresa es la responsable de determinar qué productos se van a fabricar, en qué sedes o ubicaciones y, en general, cómo van a ser fabricados.

- **Sitio** (*Site*). Una ubicación geográfica con competencias considerables de producción se identifica tradicionalmente como *site*. De hecho, la empresa en sí está constituida por un grupo de sitios o sedes. Cada sitio se planifica y organiza, a su vez, en células, líneas o unidades distribuidas en áreas. Los sitios suelen tener atribuidas unas competencias de fabricación claras.
- **Área** (*Area*). Las principales competencias de producción y ubicaciones geográficas dentro de un sitio se definen como áreas. Las áreas generalmente cuentan también con competencias de fabricación claramente definidas. Están, a su vez, formadas por elementos que llevan a cabo las diferentes funciones de fabricación. Dependiendo del tipo de proceso, estas áreas incluyen una combinación de líneas de producción (si actividad basada en operaciones discretas), de unidades de producción (si operaciones continuas) o células de proceso (si procesos por lotes).
- **Unidad de Producción** (*Production Unit, PU*). Una unidad de producción generalmente engloba todo el equipamiento que se necesita en un segmento de producción continua que opera de manera relativamente autónoma. Ejemplos de ello son, entre otros, los módulos de equipamiento, sensores o actuadores, aunque estos últimos quedan fuera del alcance de ISA95.
- **Unidad de Trabajo** (*Work Unit, WU*). Una *WU* se compone de módulos de equipos y módulos de control. Los módulos que conforman *WU* pueden ser configurados como parte de una unidad o pueden ser adquiridos temporalmente para llevar a cabo tareas específicas. Identifica la capacidad de procesamiento principal de la familia de productos. Las *WUs* son el nivel inferior del equipamiento que participa en las operaciones propias de los niveles 3 y 4 de los procesos de fabricación. Combina todos los equipos requeridos para realizar estas actividades como una agrupación de equipos independientes.
- **Módulo de Equipamiento** (*Equipment Module, EM*). Físicamente, el *EM* puede estar formado por módulos de control y otros módulos de equipamiento subordinados. Un *EM*, a su vez, puede ser parte de una unidad o de un grupo de equipos autónomos dentro de una *WU*. Un *EM* es la combinación de todos los equipos

físicos de control y procesamiento necesarios para llevar a cabo esas actividades. Suelen estar centrados en un equipo de procesamiento. Funcionalmente, el alcance de un módulo de equipo es definido por la cantidad finita de actividades que está diseñado para llevar a cabo y puede componerse de módulos de equipos subordinados y módulos de control.

- **Módulo de Control** (*Control Module, CM*). Un CM es típicamente una colección de sensores, actuadores y otros dispositivos participantes en el control, así como equipos de procesamiento asociados que, desde el punto de vista del control, operan como una entidad simple. Un CM, a su vez, puede estar formado por otros módulos de control.

Algunos ejemplos de CMs en el caso del proceso de producción de petróleo son las Bombas (*Pumps*), *Drives*, Válvulas (*Valves*), diferentes Transmisores e indicadores (*Indicating Transmitters*) como por ejemplo de presión -PIT-, de temperatura -TIT-, de nivel -LIT-, de caudal -FIP-, de presión diferencial -PDIT-, o transmisores simples (*Transmitters*) de corriente -CT-, de tensión -VT- o TT, PT, FT. La Figura 48 muestra las principales características de los *Transmitters* como

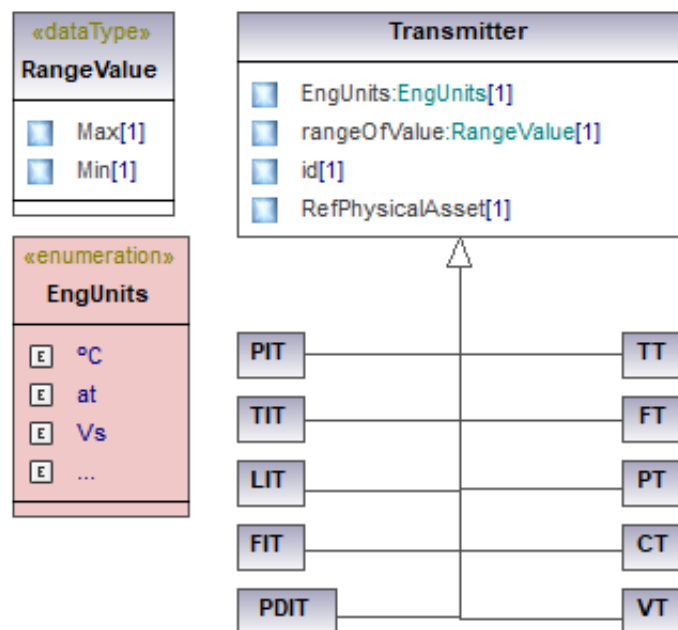


Figura 48: Caracterización de los Transmisores

son un identificador, el rango de valores, las unidades de ingeniería y el número de serie del correspondiente dispositivo físico (*physical asset*).

4.4 META-MODELO DE DISPOSITIVOS INTELIGENTES DE PLANTA (PIDM)

Los dispositivos que actúan como proveedores de datos se caracterizan por la información necesaria para acceder a los datos de proceso que proporcionan. Abarcan desde los PLCs, DCSs, sistemas empotrados, controladores industriales, sensores inteligentes hasta cualquier otro dispositivo que tenga acceso a las variables del proceso. Para definir estos dispositivos se ha optado también por utilizar el modelo de equipamiento que propone el estándar ISA95.

Las propiedades de cada dispositivo suministrador son aquellas que especifican la comunicación con los otros componentes de la red (del mismo nivel o de niveles superiores), y los datos del proceso que proporcionan. En la Figura 49 se muestran estas propiedades.

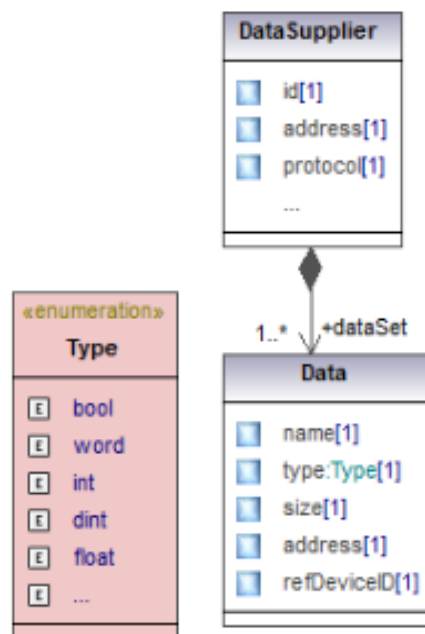


Figura 49: Meta-modelo mínimo para los dispositivos inteligentes capaces de suministrar datos

En cuanto a los aspectos de la comunicación, cada suministrador de datos se caracteriza por su dirección (*address*) y propiedades del protocolo (*protocol*) de comunicación que utilizan para suministrarlos. Por otra parte, cada dato al que se accede (*Data*) se caracteriza por un nombre (*name*), tipo (*Type*), tamaño (*size*), dirección (*address*) y la referencia al dispositivo físico (*refDeviceID*) del que lo recibe el dispositivo inteligente.

4.5 CASO DE ESTUDIO MODELADO ISA 88/95

La base de este trabajo es la arquitectura presentada en el punto 3.4 y el objetivo principal es modelar el archivo XML de configuración estática de un servidor OPC-UA desarrollado en IEC 61499 para integración vertical a nivel de planta. Específicamente usando los meta-modelos basados en ISA 88/95 propuestos en los puntos anteriores se pretende modelar el espacio de direcciones del servidor OPC-UA para ser utilizado en la industria del petróleo y gas como en el caso anterior.

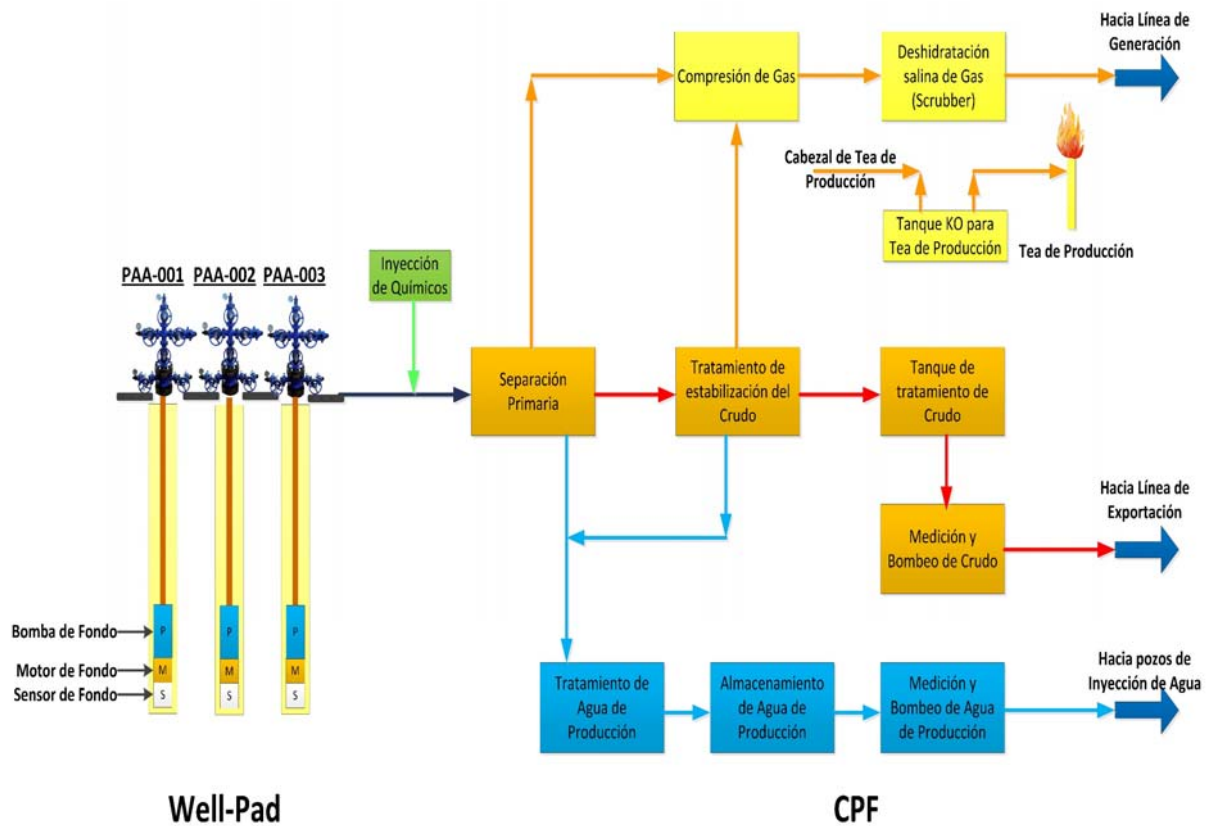


Figura 50: Proceso de Producción de Crudo

Dentro del campo de la producción de petróleo, las empresas suelen estar geográficamente distribuidas en varias regiones de un país. Por lo tanto, el acceso remoto al proceso es de vital importancia para monitorizar todo el proceso o tomar decisiones que pueden afectar a diferentes localizaciones de producción. De hecho, las compañías de producción de petróleo suelen estar compuestas de múltiples bloques de producción petrolera, denominados Well-Pad, los cuales se refieren a una superficie que contiene los pozos petroleros perforados horizontalmente para disminuir el impacto ambiental. El beneficio adicional de una plataforma de perforación es que los operadores pueden perforar múltiples pozos en un tiempo más corto de lo que podrían con sólo un pozo por localización.

Cada pozo de petróleo, tiene un sensor de fondo que recoge medidas de presión, temperatura y vibración para una identificación precisa, diagnóstico y análisis de problemas de operación de equipos y cambios en las condiciones del yacimiento.

Concretamente, el crudo se extrae del fondo de los pozos por bombeo electrosuccionable o bombeo hidráulico y una vez en la superficie se recoge este crudo en una tubería central llamado manifold de producción. Es aquí, en el manifold, donde se recoge el crudo de todos los pozos que constituyen un Well-pad (véase la Figura 50).

El proceso de extracción del crudo termina cuando el crudo se envía por bombeo al Centro de Instalaciones de Producción (en inglés Central Processing Facility o CPF) desde los Well-pads. El CPF se compone de varios equipos, uno de los más importantes es el denominado “*Separador de Crudo*”, el cual es un recipiente de presión utilizado para separar trifásicamente el crudo, en este equipo se obtienen por separado el petróleo, el agua de formación y gas. Posteriormente, el petróleo se envía a otros procesos como el de “*Almacenamiento en Tanques*” para su posterior bombeo a otras empresas de refinación o exportación. Al mismo tiempo, parte del gas se deshidrata en otro equipo denominado Scrubbing para su posterior uso en turbinas para generación eléctrica o para quemarse en tareas de producción.

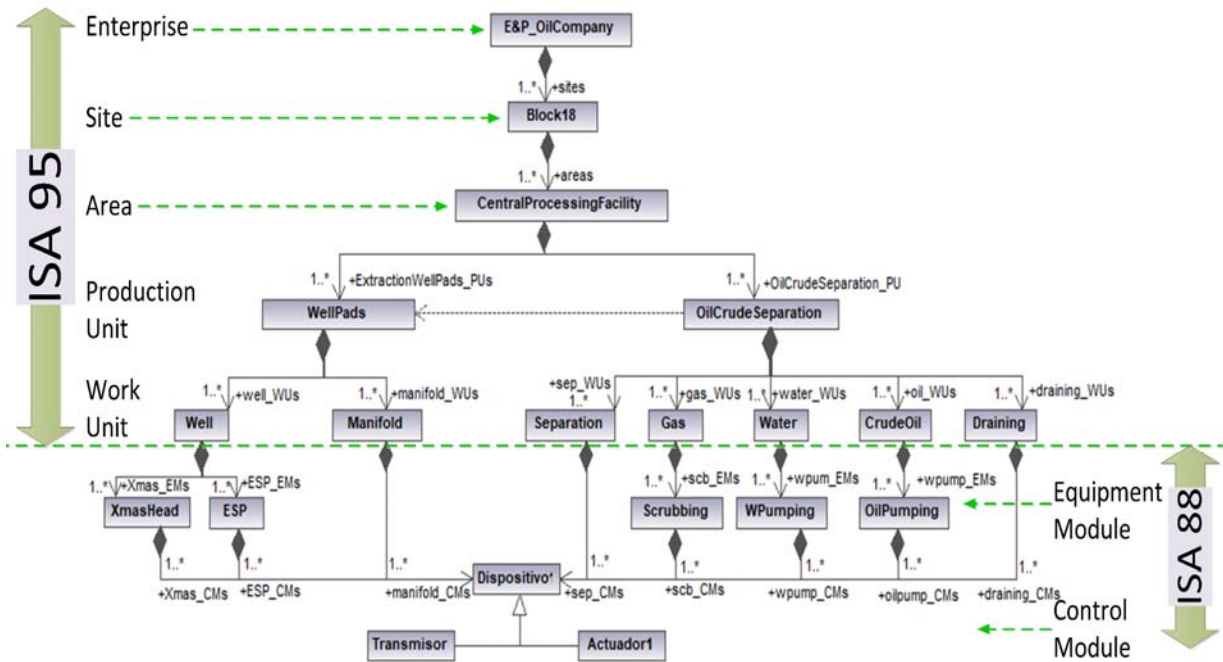


Figura 51: Meta - modelo de Topología de Planta de Petroamazona B18

El agua de formación obtenida es reinyectada al yacimiento de producción usando pozos de reinyección de agua (posee subprocessos como almacenamiento de agua y reinyección de agua). Las bombas de reinyección de agua son una de las tecnologías clave que las empresas de producción de petróleo y gas utilizan para aumentar las tasas de recuperación del petróleo; ha demostrado ser uno de los métodos más económicos para el manejo de yacimientos. Esta tecnología puede ser valiosa para ayudar a mantener la presión del yacimiento, mejorar la producción de reservas de hidrocarburos y reducir el impacto ambiental.

Al igual que en el punto 3.8, el caso de estudio se centra en Petroamazonas EP, empresa pública ecuatoriana dedicada a la exploración y producción de hidrocarburos. Está a cargo de la operación de 20 Bloques, ubicados en la cuenca Oriente de Ecuador. Con objeto de simplificar el caso de estudio, se ha modelado únicamente una parte de, en concreto, el Bloque 18. En este Bloque existen en la actualidad 4 wellpads y cada uno de ellos agrupa 30 pozos.

Aplicando los meta-modelos desarrollados para la Topología de Planta (PTM) y meta-modelo de Dispositivos Inteligentes de Planta (PIDM), podemos observar que, en el caso del campo de producción de petróleo, se considera como *Empresa* tal a la Compañía Petrolera llamada *Petroamazonas EP*. En el caso particular de esta empresa de producción de petróleo, el equivalente al concepto estándar de *Sitio* sería el *Bloque (Block)*. El concepto de *área* en este tipo de proceso podría presentarse por el *Centro de Facilidades de Producción (CPF)*. Por tanto, cada Bloque de una Compañía de producción y explotación de petróleo estará formado por un conjunto de CPFs, equivalente al concepto estándar de *área*. Unidades de producción pueden identificarse dos: la PU de *Well Pad (extracción de crudo)* y la PU de *Separación del Crudo* (ver Figura 51).

Así, cada PU de extracción del wellpad se caracteriza por dos tipos de WU: (i) *pozos para extracción de crudo o reinyección de agua de formación* y (ii) el conjunto de *tuberías (manifold)* que transportan dicho fluido a la PU de separación del crudo. Del mismo modo, la PU de separación del crudo está formada por cinco WUs: (i) *Separación*, (ii) *Gas*, (iii) *Agua*, (iv) *Petróleo* y (v) *Drenaje* (Ver Figura 51).

Para este caso de estudio en particular, se ha identificado un conjunto de EMs. Tal es el caso de la WU pozo, que está formada por dos EMs: (1) *la cabecera del pozo (Xmas-Head)* y (2) el sistema de *bombeo electrosumergible (ESP)*.

El ejemplo tiene por objeto definir una parte del proceso, en concreto se modelará el equipo que es corazón del sistema de producción, llamado separador de crudo horizontal *V1401*. Este tipo de separador se denomina separador de agua libre porque su uso principal es retirar el agua de formación que podría causar problemas como corrosión y formación de hidratos o emulsiones compactas que son difíciles de descomponer. A un separador de agua libre se le denomina separador trifásico, porque puede separar gas, petróleo o agua libre. El separador de agua libre se abrevia *FWKO*, por sus siglas en inglés free water knockout.

Las variables que están implícitas en el proceso son i) Temperatura se debe controlar para mantener la viscosidad del crudo en un punto óptimo y que no afecte al proceso,

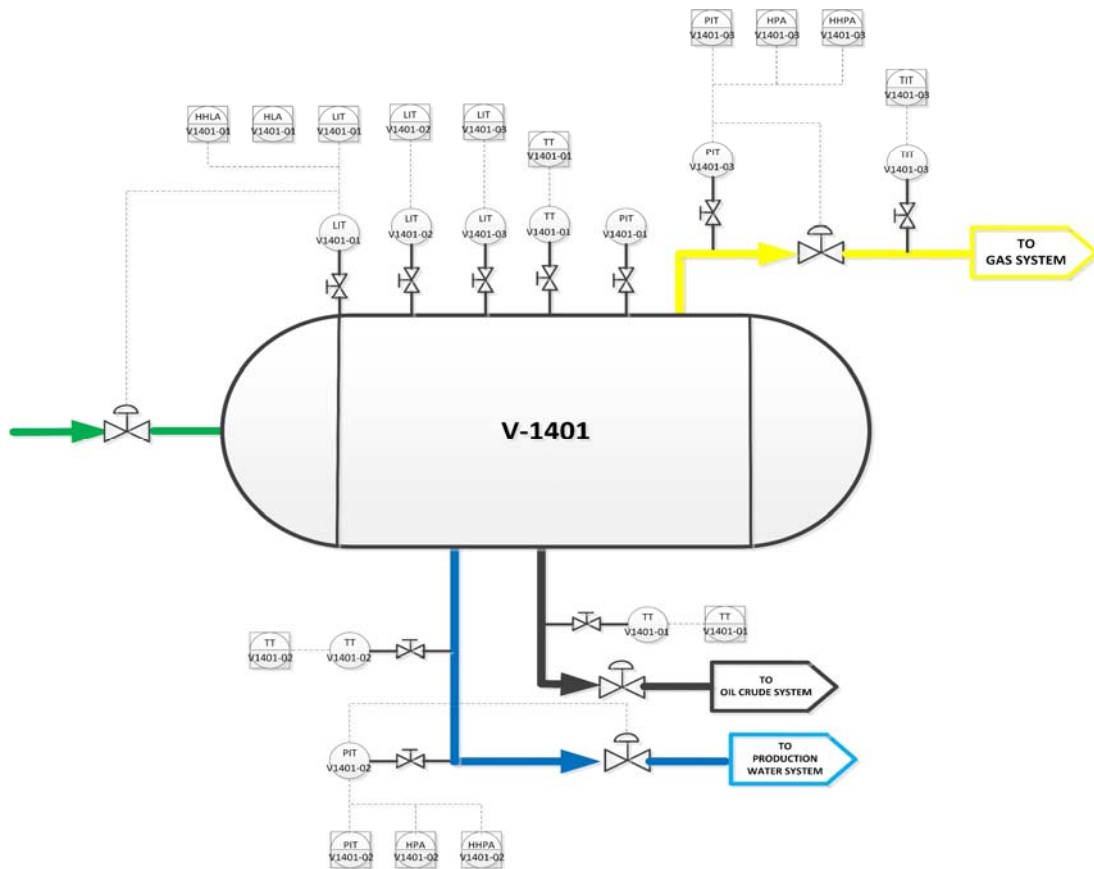


Figura 52: Instrumentación separador de crudo V1401

en el separador se tiene 3 transmisores de temperatura: *TIT-V1401-1* para temperatura de crudo, *TIT-V1401-2* para la temperatura del agua de formación y *TIT-V1401-3* para la línea de gas. ii) Nivel tiene consecuencias en el causa efecto del proceso llevando al sistema a paro general (shutdown) si se tocan los límites de alto-alto o bajo-bajo. El separador tiene 4 transmisores de nivel, *Cy LIT-V1401-2* para medición del nivel de crudo, *LIT-V1401-3* y *LIT-V1401-4* para medir el nivel de agua de formación. iii) Flujo esta variable que controla una válvula que permite distribuir el flujo en los trenes de entrada del separador, el transmisor para este control es el *FT-V1401-1*. iv) Presión, evita que pase crudo a la línea de gas y tiene efectos en el causa efecto del sistema llevándolo a paro general (shutdown) por alta presión. Los transmisores son *PIT-V1401-1* para presión de crudo, *PIT-V1401-2* para presión de agua de formación y *PIT-V1401-3* para presión de gas que pasa al siguiente proceso de secado de gas en el tanque de scrubbing.

OilCompanyEnterprise					
= Name PetroamazonasEP					
Block					
= id Block18					
CentralProductionFacilities					
OilSeparation					
= id V1401					
Separator					
LIT (4)					
	= id	= EngUnits	= min	= max	= refPhysicalAsset
1	LIT-V1401-1	ft	0	10	HART01
2	LIT-V1401-2	ft	0	7	HART02
3	LIT-V1401-3	ft	0	5	HART03
4	LIT-V1401-4	ft	0	3	HART04
TIT (3)					
	= id	= EngUnits	= min	= max	= refPhysicalAsset
1	TIT-V1401-1	°C	0	800	HART05
2	TIT-V1401-2	°C	0	600	HART06
3	TIT-V1401-3	°C	0	600	HART07
PIT (3)					
	= id	= EngUnits	= min	= max	= refPhysicalAsset
1	PIT-V1401-1	PSI	0	600	HART08
2	PIT-V1401-2	PSI	0	800	HART09
3	PIT-V1401-3	PSI	0	600	HART10
SDV id=SDV-V1401-3 EngUnits=open/close min=0 max=1 refPhysicalAsset=... FT id=FT-V1401-3 EngUnits=barrels min=0 max=1200 refPhysicalAsset=MO...					
WellPad					

Figura 53: Ejemplo de modelo de Topología de Planta

Asimismo, cuenta con una válvula de emergencia a la entrada del separador, la *SDV-V1401-1*. (ver Figura 52)

Toda esta información que describe la topología de la planta y los dispositivos inteligentes se guarda en documentos con notación de lenguaje de marcado (*Markup Language, ML*). Estos documentos siguen su correspondiente *schema XML* -PTMB18.xsd y PIDMB18.xsd respectivamente- que son implementaciones basadas en XML de los metamodelos presentados en el apartado anterior. En la Figura 53 se puede observar una parte del modelo PTM aplicado al caso de estudio, y una muestra de la parte correspondiente al modelo PIDM en la Figura 54.

PlantIntelligentDeviceModel								
Data Supplier (2)								
id	protocol	DATA						
1	DELTAVMseries	HART	DATA (10)					
			name	type	size	address	refDeviceID	
			1	crude_level_up	WORD	16	CH01	LIT-V1401-1
			2	crude_level_down	WORD	16	CH02	LIT-V1401-2
			3	water_level_up	WORD	16	CH03	LIT-V1401-3
			4	water_level_down	WORD	16	CH04	LIT-V1401-4
			5	crude_temperature	WORD	16	CH05	TIT-V1401-1
			6	water_temperature	WORD	16	CH06	TIT-V1401-2
			7	gas_temperature	WORD	16	CH07	TIT-V1401-3
			8	crude_level	WORD	16	CH08	LIT-V1401-1
			9	water_level	WORD	16	CH09	LIT-V1401-2
			10	gas_level	WORD	16	CH10	LIT-V1401-3
2	CONTROLOGIX5000	MODBUS	DATA (3)					
			name	type	size	address	refDeviceID	
			1					
			2	SDV_Emergency	BOOL	1	P01_DEV01	SDV-V1401-1
			3	crude_production	WORD	16	P01_DEV01	LIT-V1401-1

Figura 54: Ejemplo de modelo de Dispositivos

4.6 CREACIÓN DE LA HERRAMIENTA PARA GENERACIÓN DE APLICACIONES DISTRIBUIDAS

Actualmente existen varias herramientas de software como: ISaGraf[149], ntxControl [150] and 4DIAC -IDE [151] donde se pueden desarrollar *aplicaciones de control distribuidas* conforme al estándar. Al aplicar la metodología desarrollada en el punto 3.5.3 para la implementación de aplicaciones de control distribuida en las herramientas software, se llega a la conclusión de que este tipo de herramientas no son utilizadas en la industria, especialmente en el sector del petróleo y gas, debido a su complejidad, desconocimiento o falta de personal cualificado para interactuar con ellas.

El reto y objetivo de esta parte del capítulo es presentar un Entorno de Desarrollado Integrado (IDE) de código abierto como una herramienta de referencia industrial bajo los lineamientos de la norma IEC 61499 y su aplicación en diversos dominios de la automatización industrial para desarrollar Componentes de Información (ICs) utilizando la arquitectura CPPS presentado en los capítulos anteriores y usando OPC-UA como protocolo de comunicación, permitiendo a los ingenieros que trabajan en la industria del petróleo y gas incorporar esta norma en sus instalaciones.

A continuación, se mencionan trabajos relacionados referentes a este tema.

4.6.1 Trabajos Relacionados

Las posibilidades de implementación de la norma IEC 61499 en dispositivos industriales están actualmente limitadas debido al bajo número de plataformas capaces de ejecutar directamente los FBs en dispositivos industriales. Esto se puede ver en proyectos como: plataforma Elsis Netmaster II [152] y la plataforma Tait Control Systems Intelligence [153]. La herramienta de ingeniería TORERO [106] abarcó todo el ciclo de vida de desarrollo de sistemas de control distribuidos utilizando la norma IEC 61499. Otra aplicación desarrollada es el sistema de soporte de ingeniería llamado CORFU [154] que intentó establecer un marco de software inspirado en modelos de aplicaciones de control utilizando IEC 61499 y Lenguaje de Modelado Unificado (UML). Pero estos trabajos no han sido suficientes debido a que los investigadores deben desarrollar nuevos enfoques de software y arquitecturas para posibilitar la implementación del estándar IEC 61499 en el entorno industrial real.

Incluso cuando los investigadores han desarrollado una gran cantidad de aplicaciones de control basadas en la norma IEC 61499 en los últimos años, la mayoría se limitó a controlar sistemas industriales de menor escala y no pudieron resolver problemas de confiabilidad resultantes de las perturbaciones de estos sistemas. Hussain et al. [88] muestran cómo las aplicaciones distribuidas, reconfigurables y flexibles pueden ser modeladas usando IEC 61499. Para ello, utilizan una plataforma de software llamada NETMASTER, desarrollada en Java y cuyo núcleo es la programación basada en C++. Otra investigación importante es la desarrollada por Jain et al. [85] donde se presenta un marco para prototipado rápido y simulación de sistemas llamado EMBENCH, que se utiliza para controlar la maquinaria en diferentes niveles (control numérico, fabricación flexible, etc.) utilizando IEC 61499 como herramienta para la segmentación y reutilización de servicios de control.

En conclusión, las arquitecturas CPPS y el análisis inteligente de Grandes Volúmenes de Datos (Big Data) es una solución factible para proporcionar predicción y eficiencia

en la personalización del producto y mejorar la producción en una planta industrial. Sin embargo, no se analizan problemas como la reutilización de software, la comunicación de plantas o su uso en plataformas de hardware para CPPS.

Hoy en día, hay aplicaciones de CPPS en empresas industriales reales. Uno de los principales desafíos es diseñar diferentes metodologías de modelado que permitan gestionar sistemas complejos de producción industrial, donde es probable que su complejidad aumente en el futuro. En este sentido, para sistemas de automatización distribuidos con procesos físicos integrados es necesario contar con metodologías de diseño basadas en técnicas de modelado como las que se apoyan en la norma IEC 61499. Desde el punto de vista del diseño, se necesita lo siguiente: modelos de software para las diversas plataformas existentes, modelos para la semántica de ejecución de las aplicaciones y comunicación industrial utilizando nuevos protocolos como OPC-UA o MQTT. Esto se presenta en el trabajo de Dai et al [96] donde el objetivo principal es proponer modelos de control integrados y software de comunicación para el diseño de sistemas distribuidos en aplicaciones Ciber-Físicas utilizando la norma IEC 61499. Sin embargo, el uso de nuevos estándares de comunicación como OPC-UA no se presenta en esta investigación.

Ferrarini et al. [97], [98] promueve un método de modelado para procesos industriales bajo la norma IEC 61499 basado en el uso de componentes de automatización industrial con una distribución similar a la estructura de la planta o de las máquinas usadas para la división de los procesos en las fábricas. En esta línea, Thramboulidis [99] aplica técnicas de modelado utilizando la norma IEC 61499 y presenta la arquitectura MIM que se basa y adapta conceptos de MDA (Model Driven Architecture) proponiendo un entorno para modelar componentes mecatrónicos que participan en procesos industriales.

Kozar et al [56] introduce un enfoque que une la plataforma 4DIAC con el protocolo de comunicación OPC-UA y tiene como objetivo utilizar conceptos y modelos dados por este protocolo para el diseño de sistemas distribuidos. Además, integra el descubrimiento de la estructura de la aplicación basado en el modelo de información de OPC-UA y por lo tanto no requiere ninguna aportación adicional del desarrollador del sistema. Sin embargo, esta tesis va más allá de los trabajos anteriormente descritos, como se ha visto

los servidores OPC-UA descritos permiten la integración de este protocolo con normas de comunicación industrial usadas en la industria como Modbus TCP, TCP/IP, etc para acceder a los datos de campo a nivel de planta en una arquitectura CPPS de bajo coste, es por esto que necesita de una herramienta de fácil uso que permita la implementación real en plantas industriales como la del petróleo y gas.

4.7 DESARROLLO DE HERRAMIENTA DE INGENIERÍA

Como se ha comentado anteriormente, el objetivo es desarrollar una herramienta de ingeniería de programación sencilla para crear componentes de información basados en la norma IEC 61499 y OPC-UA para su uso en la industria de gas y petróleo.

4.7.1 Perfil de Conformidad para verificación de la factibilidad de software basados en IEC 61499

En los primeros años de la norma IEC 61499 se definió un perfil de cumplimiento que originalmente fue desarrollado bajo la forma de un acuerdo técnico por parte de miembros del Consorcio HMS (Holonc Manufacturing Systems) para guiar el desarrollo de herramientas software y verificar la interoperabilidad entre vendedores de herramientas basados en IEC 61499. El Perfil de Cumplimiento para la Demostración de Factibilidad (CPFD del inglés Compliance Profile for Feasibility Demonstrations) se puede llamar un éxito ya que este perfil sigue siendo de hecho estándar cuando se utiliza IEC 61499.

Este perfil permite cumplir con las condiciones de portabilidad, interoperabilidad y configurabilidad para las herramientas software, las cuales, deben ser capaces de entender todos los elementos de la biblioteca de FBs e SIFBs tanto de comunicación como de control soportados por los dispositivos hardware a ser usados. Este perfil permite interpretar la sintaxis y generar correctamente el programa que va a ser descargado de acuerdo con la definición DTD XML dada en el Anexo A de la norma IEC 61499. Esto permite una correcta comunicación entre la herramienta de programación y los dispositivos hardware que tienen un runtime IEC 61499 asociado a ellos.

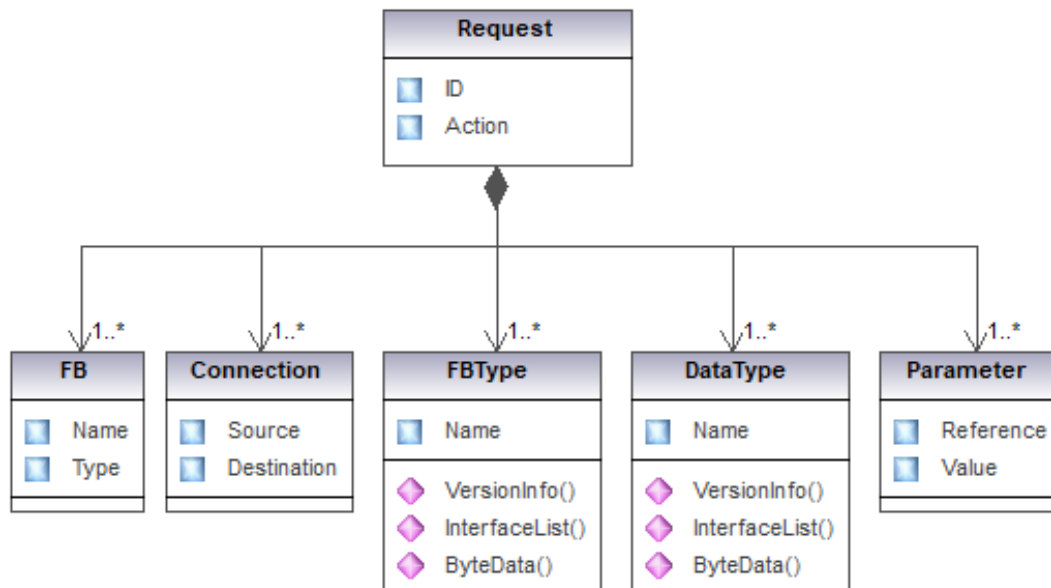


Figura 55: Metamodelo servicio REQUEST

Esta sección del capítulo presenta una herramienta de software basado en modelos que permite crear un archivo de configuración que puede ser descargado en runtimes IEC 61499 para el diseño de componentes de información basados en servidores OPC-UA desarrollados en los capítulos anteriores. Todos los runtime basados en la norma IEC 61499 deben entender este tipo de archivos de configuración ya que cumple con los requisitos del CPF. Esta herramienta permite crear dispositivos software basados en Componentes de Información y administrar los recursos del hardware a usarse llamando a los comandos de configuración dados por la norma IEC 61499. A continuación, se explican los comandos diseñados para esta herramienta software.

El CPF da a las herramientas software el servicio de gestión REQUEST, que permite comunicarse con los dispositivos hardware o software que tengan integrados runtimes IEC 61499. El diagrama de clases UML del meta-modelo correspondiente a este servicio se representa en la Figura 55, en donde se detalla los elementos de modelado necesarios para realizar una solicitud. La clase REQUEST tiene atributos como *ID*, que muestra un identificador único para la transacción de petición/respuesta y *ACTION* es la operación solicitada que se va a realizar. Las clases relacionadas son: *FB* que muestra un bloque de función o una instancia de recurso tal como se define en IEC 61499-1, tiene

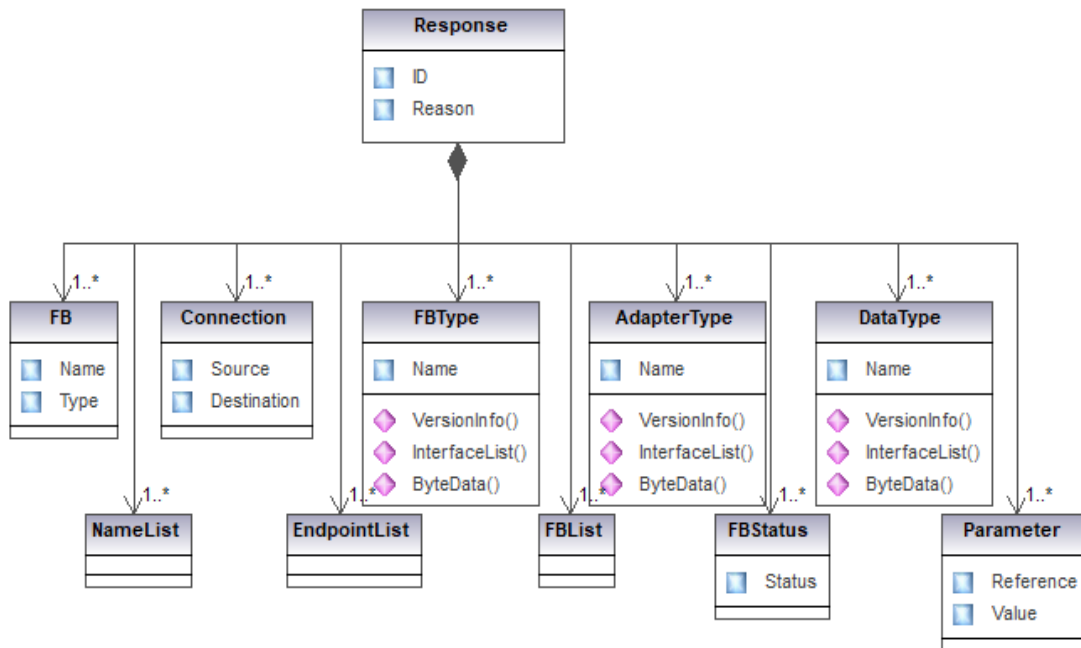


Figura 56: Metamodelo servicio RESPONSE

atributos tales como Nombre (*Name*) y Tipo (*Type*) que indica el nombre y tipo del FB o recurso respectivamente; *CONNECTION* que indica una conexión de evento, conexión de datos o conexión de adaptador; *FBTYPE* y *FBTypeDeclaration* como se describe en IEC 61499-1-C.1.1. La clase *DATATYPE* indica el tipo de datos de las entradas o salidas de los FBs. Finalmente, la clase *PARAMETER* que tiene parámetros por referencia, por valor o ambos, tiene los atributos Referencia (*Reference*) y Valor (*Value*).

Adicionalmente el CPFDD da a los recursos hardware o software donde se ejecuta el runtime IEC 61499 el servicio RESPONSE de esta manera la herramienta software de configuración interpreta si los comandos de configuración han sido realizados con éxito en el runtime. En la Figura 56 se muestra el meta-modelo de la clase RESPONSE de IEC 61499. En este meta-modelo se incluyen las 5 clases descritas anteriormente del servicio REQUEST que son: *FB*, *CONNECTION*, *FBType*, *DATATYPE* y *PARAMETER*. Además de estas clases posee la clase *NAMELIST*, donde se muestra una lista de los tipos de FBs o nombres de tipo de datos, y la clase *ENDPOINTLIST* que indica una lista de las fuentes o destinos de todas las conexiones de los FBs. La clase *FBLIST* tiene una lista (puede estar vacía) de nombres de todas las instancias administradas del tipo especificado o tipos

de FBs. Por último, la clase *FBSTATUS* indica cualquiera de los estados posibles para el FB: *INVALID_STATE*, *UNSUPPORTED*, *INVALID_OPERATION*, entre otros.

Para realizar la comunicación entre la herramienta de software y los recursos de hardware de acuerdo al CPFD se realiza usando los protocolos basados en el Protocolo de Internet (IP), es decir, Protocolo de datagramas de usuario (UDP) y Protocolo de control de Transmisión (TCP). Estos protocolos fueron elegidos inicialmente por la norma IEC 61499 debido al creciente uso de protocolos basados en IP en aplicaciones de automatización industrial y la disponibilidad general de estos protocolos en computadoras personales, aunque actualmente es una desventaja ya que no considera nuevos protocolos basados en Ethernet que están dirigidos a aplicaciones en tiempo real.

4.7.2 Diseño de software para la creación de archivos de configuración basados en IEC 61499

Después de desarrollar el meta-modelo del archivo de configuración para crear aplicaciones que se pueden descargar en runtimes de acuerdo con la norma IEC 61499, necesitamos diseñar una herramienta de software capaz de transformar el meta-modelo en un archivo XML que contiene información para crear FB con Componentes de Información utilizando servidores OPC-UA.

Esta herramienta de ingeniería sigue como referencia el meta-modelo del Perfil de Cumplimiento para Demostraciones de Factibilidad (IEC 61499-4) explicado en la sección anterior. Es una herramienta sencilla y fácil de usar que permite crear servidores estáticos y dinámicos OPC-UA bajo la norma IEC 61499.

La herramienta de software es una aplicación que lleva a cabo los mecanismos de composición de una aplicación de control distribuida una vez especificados los mapeos a dispositivos. Para ello, hace uso de una librería que contiene la información de los SIFBs de OPC-UA a partir de la que genera la aplicación completa.

El panorama general se ilustra en la Figura 57. Representa los elementos que necesita la herramienta para llevar a cabo los mecanismos de composición. Es una aplicación

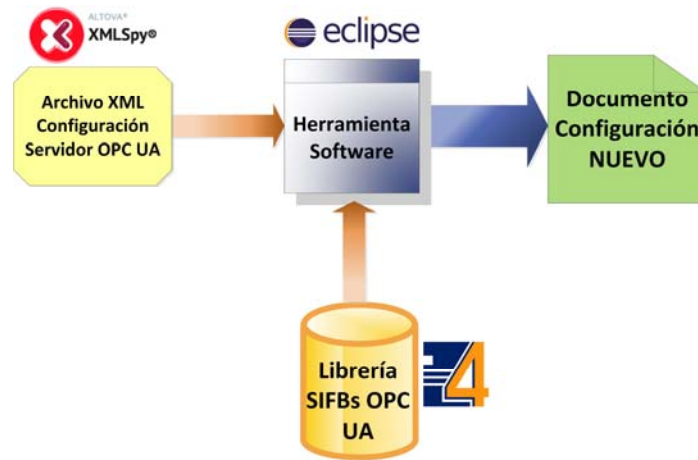


Figura 57: Modelo General Herramienta Software

independiente de la plataforma la cuál toma como entrada un modelo de sistema basado en XML explicado en el capítulo anterior y lo transforma a una aplicación completa incluyendo los SIFBs de servidores OPC-UA correspondientes de manera automática, generando un nuevo documento de sistema.

Las entradas de la herramienta son:

1. El documento de configuración estática para integración vertical de planta, que contiene información en XML de la aplicación, los dispositivos, los recursos y redes.
2. La librería de SIFBs, que contiene la información de los SIFBs de servidores y clientes para comunicación con protocolo OPC-UA.

La herramienta de software es el responsable de la configuración de toda la información requerida, así como de garantizar el comportamiento correcto de la aplicación de control distribuida.

Esta herramienta principalmente se encarga de incluir los SIFBs, las conexiones y los datos de configuración necesarios en los dispositivos correspondientes para la distribución de la aplicación de forma automática.

Las ventajas que aporta esta herramienta son:

- No tiene por qué formar parte del sistema de desarrollo donde se crea la aplicación y el sistema de control distribuido.
- Es independiente del sistema de desarrollo.
- Es modular.
- Fácil de extender.
- Lee la información en formato XML.
- Inserta automáticamente el código correspondiente de los SIFBs de comunicaciones.
- Crea automáticamente las conexiones entre los FB de comunicaciones y los FBs distribuidos en los dispositivos.
- Está bajo el CPFD de la norma IEC 61499-4.

Como se ha mencionado anteriormente en el desarrollo del documento de sistema, es necesario contar con una serie de herramientas software que permitan manipular esta información. Para que la herramienta realice estas acciones, se han utilizado las herramientas software que se muestran en la Figura 57. 4DIAC-IDE se ha utilizado para el diseño y desarrollo de los SIFBs de comunicación, 4DIAC-FORTE como runtime descargado en dispositivos. XMLSpy se ha utilizado para el diseño y desarrollo del archivo de configuración estático modelado bajo las normas ISA 88/95 y, por último, Eclipse se ha utilizado para el desarrollo de la herramienta que se ejecuta bajo JAVA. El nuevo documento de sistema creado es compatible con el perfil CPFD.

El meta-modelo del software diseñado se muestra en la Figura 58. La clase principal es FB_OPCUAServer que permite crear dos tipos de servidores. El primer tipo se llama SERVER_0 y sólo permite una configuración inicial y estática del servidor OPC-UA. La configuración estática utiliza un archivo XML que incluye todos los parámetros esenciales para el servidor OPC-UA, como dirección URL, nombre del proveedor, nombre del servidor, versión, etc. Además, este archivo de configuración indica el espacio de direcciones modelado en ISA88/95 con sus tipos de nodo e instancias de nodo, entre otros. El segundo tipo se llama SERVER_1, permite crear componentes de información dinámicamente. El formato de SIFB utilizado se presentó en el capítulo 3.

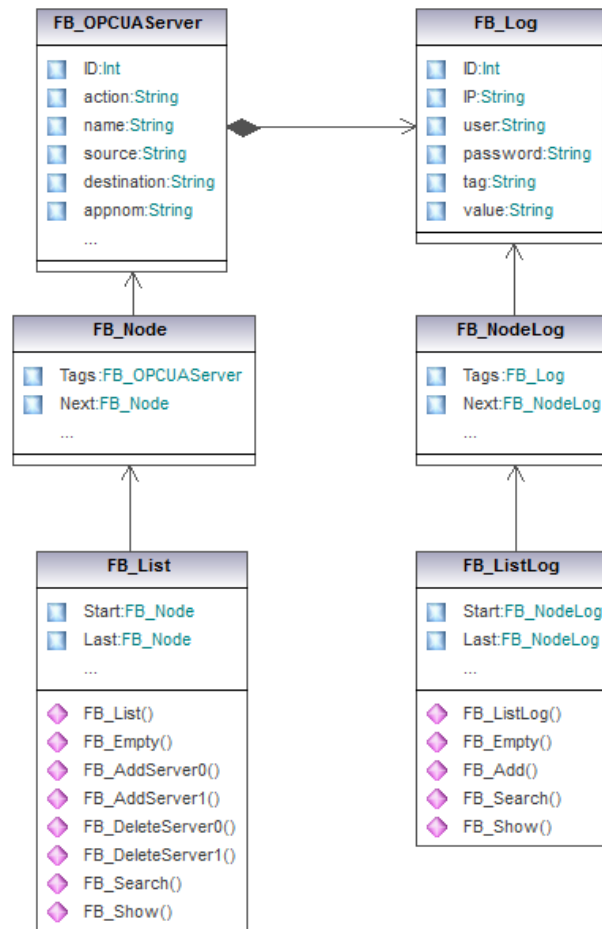


Figura 58: Metamodelo de la Herramienta Software

El servidor de tipo SERVER_1 tiene una entrada para leer variables de proceso y se le ha dado la capacidad de almacenar los datos en una Base de Datos basada en SQL (por sus siglas en inglés Structured Query Language; en español lenguaje de consulta estructurada). Para lograr esta función se ha desarrollado e implementado un SIFB llamado LOG DATA. Este FB utiliza las bibliotecas de PostgreSQL para integrar IEC 61499 con bases de datos. Se trata de bibliotecas de código abierto y orientadas a objetos y de distribución gratuita. Una característica interesante de PostgreSQL es el control de concurrencia multiversión o MVCC, que permite realizar transacciones eventualmente consistentes que aportan grandes ventajas en el rendimiento. Tiene interfaces de programación nativas para C/C++, Java, Net, Perl, Python, Ruby, Tcl, ODBC.

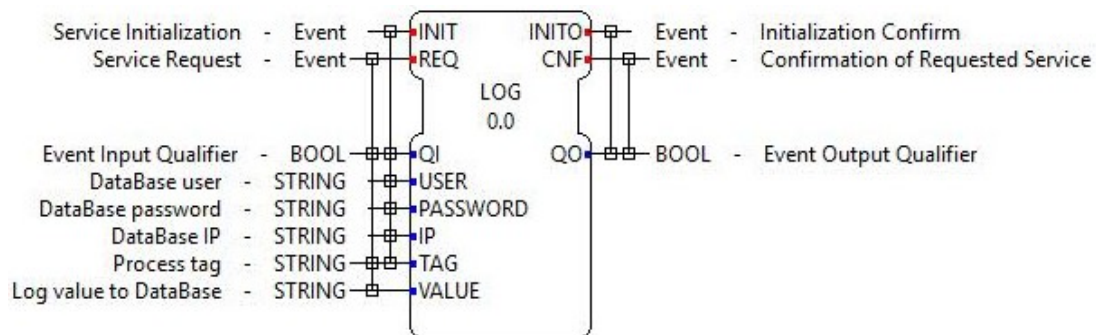


Figura 59: SIFB LOG DATA

EL SIFB LOG_DATA posee eventos comunes como INIT, REQ, INITO y CNF (Figura 59), así como los siguientes datos de entrada y salida:

- USUARIO (STRING): Los nombres de usuario de la base de datos son globales en una instalación de clúster de base de datos (y no en una base de datos individual).
- PASSWORD (STRING): Introduce la contraseña del usuario seleccionado en la base de datos.
- IP (STRING): Dirección IP para crear el socket de comunicación con la base de datos.
- TAG (STRING): Variable de proceso para iniciar sesión en la base de datos.
- VALUE (STRING): Valor de los datos a escribir. Una vez más, el tipo de este parámetro es ANY con el fin de aumentar la generalidad.

Continuando con la explicación de la herramienta software, el servidor SERVER_1 es el único capaz de conectarse con el SIFB LOG_DATA. Para la creación de cualquier tipo de servidor descrito en las secciones anteriores, se necesitan dos clases. La primera clase es FB_Node permite crear los nodos necesarios para hacer posible la comunicación entre FORTE y la pila de comunicación del protocolo OPC-UA. Finalmente, la segunda clase FB_List tiene todas las funciones para borrar servidores, buscar Información Componentes dentro del servidor, etc, (Ver Figura 58)

Para crear el FB para registro de datos (FB LOG_DATA), la clase principal necesita valores como IP, usuario, contraseña, etiqueta y el valor de proceso que se almacenará en

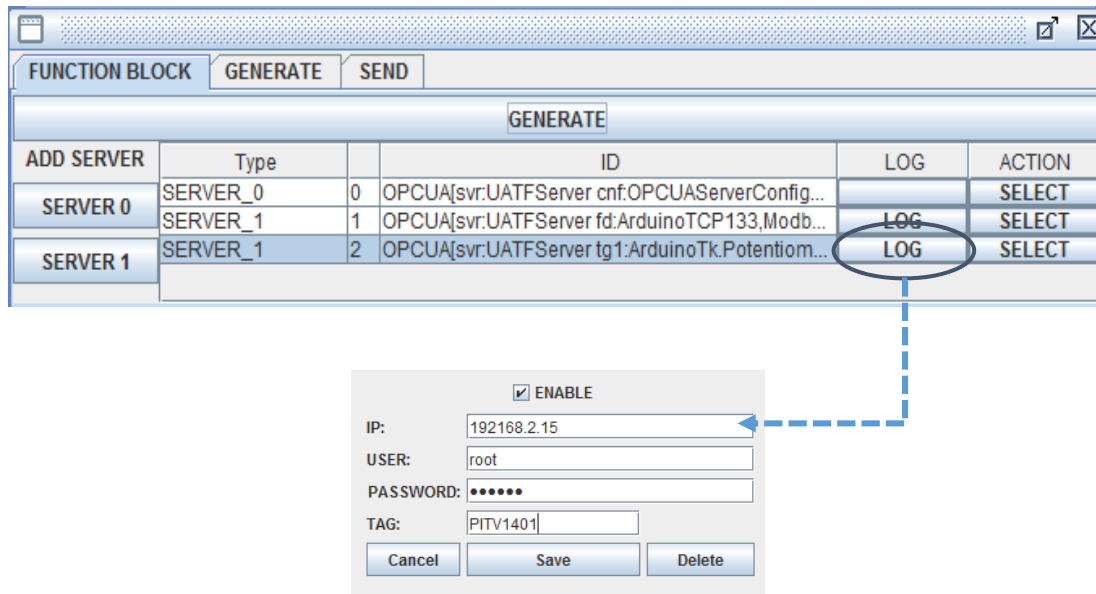


Figura 60: Función FUNCTION BLOCK de Herramienta Software

la base de datos PostgreSQL. Para crear el servidor OPC-UA esta clase tiene dos clases similares a la descripción anterior que son: FB_NodeLog y FB_NodeList que proporciona las funciones de agregar un nuevo FB al registro de datos, eliminar, buscar, etc

A continuación, se muestra la descripción de cada una de las funciones de la herramienta software. Esta herramienta de ingeniería se ha programado usando en el software Java, tal como se presenta en la Figura 60, sigue al meta-modelo previamente explicado.

Esta herramienta tiene tres funciones básicas:

- **FUNCTION BLOCK**: Permite la creación de servidores OPC-UA de los tipos SERVER_0 y SERVER_1, el único parámetro que se debe ingresar es ID usa el formato que se describe en la sección 3.7. Adicionalmente, tiene la opción de DATA LOG, que hace posible comunicarse con una base de datos relacional para registrar las variables de proceso. Esta función permite también eliminar los servidores creados. (ver Figura 60)
- **CREATE**: Una vez ingresado todos los parámetros de los servidores OPC-UA para crear Componentes de Información, esta función genera el archivo XML que será descargado en runtimes de IEC 61499 el cual cumple el CPFD dado por esta

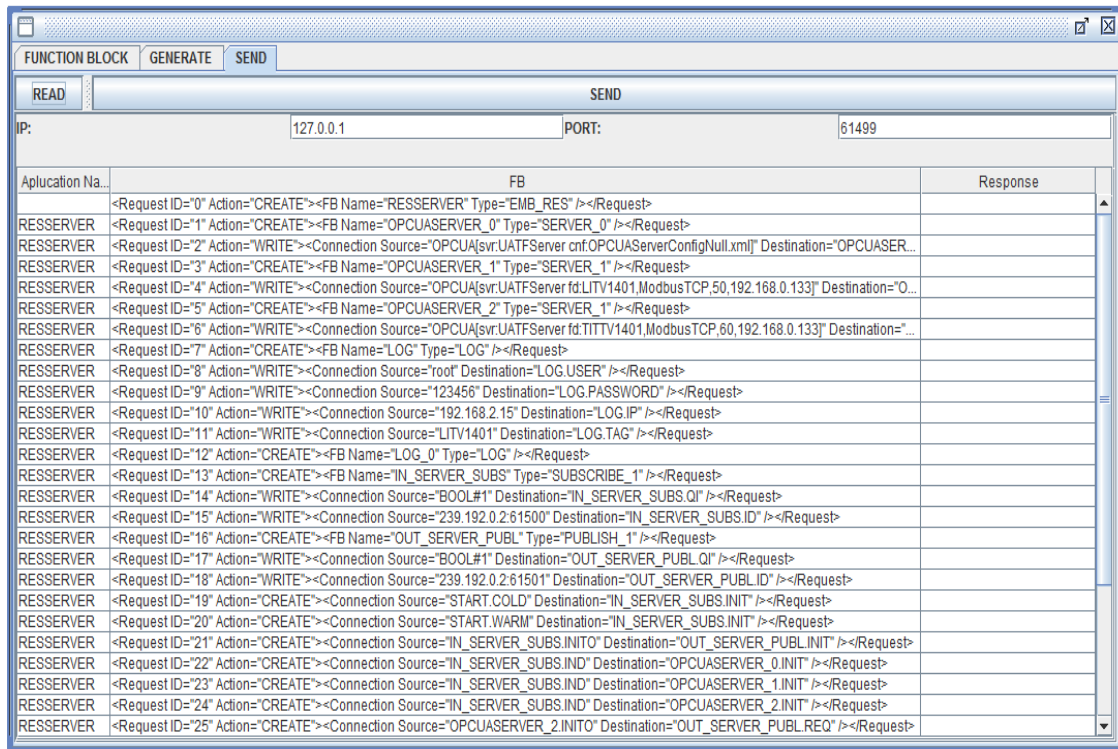


Figura 61: Función SEND de Herramienta Software

norma. Está compuesto de dos partes bien definidas, la primera crea los SIFBs de servidores OPC-UA y la segunda parte implementa todas las conexiones de eventos y/o datos de fuente y destino de la aplicación que se creará en runtimes IEC 61499. Este archivo XML al ser descargado permite que los runtimes lean las instrucciones y creen recursos IEC 61499 para adquisición y comunicación de datos.

- **SEND:** Permite descargar el archivo XML generado en el dispositivo CPPS; en esta opción se necesita una dirección IP y el puerto de comunicación para iniciar la descarga debido a que las aplicaciones IEC 61499 utiliza el protocolo TCP/IP para esta opción. Usa el CPFD y siempre que se envía una línea del archivo XML se recibe una respuesta del runtime indicando uno de los estados posibles de los recursos creados: Estado inválido, recurso inválido o recurso no creado. (ver Figura 61).

4.8 CASO ESTUDIO HERRAMIENTA DE INGENIERÍA SOFTWARE

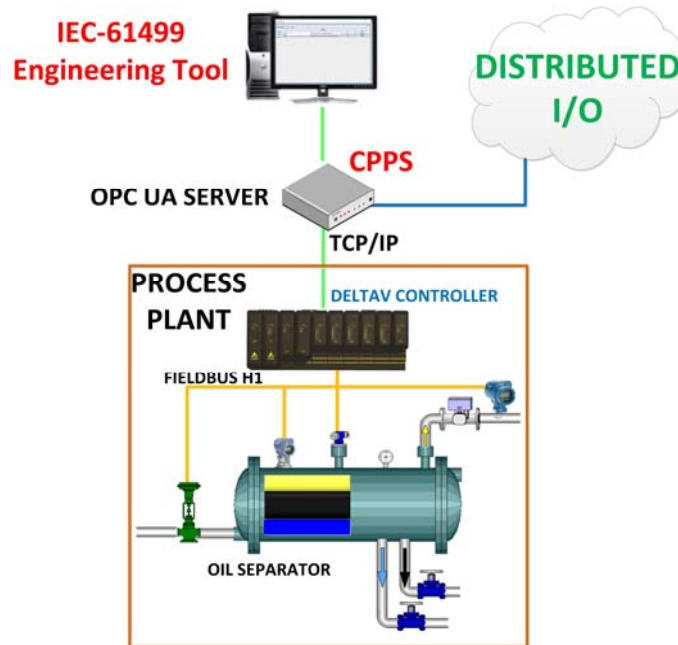


Figura 62: Arquitectura de comunicación Separador de Crudo V1401

El proceso para el caso de estudio es el explicado en el punto 4.5. El objetivo principal es utilizar la herramienta de ingeniería para crear nuevas variables en servidores OPC-UA de manera dinámica, las variables (tags) son todos los instrumentos instalados en el separador de crudo de petróleo V-1401. La arquitectura del sistema de comunicación industrial se muestra en la Figura 62. Se observa en esta arquitectura que un dispositivo CPPS está conectado a la red TCP/IP, en este dispositivo se embebe un servidor OPC UA, el cual puede acceder a los datos de proceso dados por el controlador DELTAV serie M utilizando la pila de comunicación implementada en el servidor OPC-UA para la norma IEC 61499. Los pasos para usar esta herramienta de ingeniería son: i) Primero, usar la función FUNCTION_BLOCK para crear servidores OPC-UA estáticos (SERVER_0) o dinámicos (SERVER_1) usando la codificación dada en el punto 3.7.1 y 3.7.2, ii) un vez creado los Componentes de Información se selecciona la opción GENERATE para crear el archivo de configuración XML que cumple con todas las especificaciones del CPFD

```

;<Request ID="0" Action="CREATE"><FB Name="RESSERVER" Type="EMB_RES" /></Request>
RESSERVER;<Request ID="1" Action="CREATE"><FB Name="OPCUASERVER_0" Type="SERVER_0" /></Request>
RESSERVER;<Request ID="2" Action="WRITE"><Connection Source="null" Destination="OPCUASERVER_0.ID" /></Request>
RESSERVER;<Request ID="3" Action="CREATE"><FB Name="OPCUASERVER_1" Type="SERVER_1" /></Request>
RESSERVER;<Request ID="4" Action="WRITE"><Connection Source="null" Destination="OPCUASERVER_1.ID" /></Request>
RESSERVER;<Request ID="7" Action="CREATE"><FB Name="LOG" Type="LOG" /></Request>
RESSERVER;<Request ID="8" Action="WRITE"><Connection Source="root" Destination="LOG.USER" /></Request>
RESSERVER;<Request ID="9" Action="WRITE"><Connection Source="123456" Destination="LOG.PASSWORD" /></Request>
RESSERVER;<Request ID="10" Action="WRITE"><Connection Source="192.168.0.124" Destination="LOG.IP" /></Request>
RESSERVER;<Request ID="11" Action="WRITE"><Connection Source="LIT-V1401-01" Destination="LOG.TAG" /></Request>
RESSERVER;<Request ID="12" Action="CREATE"><FB Name="LOG_0" Type="LOG" /></Request>
RESSERVER;<Request ID="13" Action="CREATE"><FB Name="IN_SERVER_SUBS" Type="SUBSCRIBE_1" /></Request>
RESSERVER;<Request ID="19" Action="CREATE"><Connection Source="START.COLD" Destination="IN_SERVER_SUBS.INIT" /></Request>
RESSERVER;<Request ID="20" Action="CREATE"><Connection Source="START.WARM" Destination="IN_SERVER_SUBS.INIT" /></Request>
RESSERVER;<Request ID="21" Action="CREATE"><Connection Source="IN_SERVER_SUBS.INITO" Destination="OUT_SERVER_PUBL.INIT" /></Request>
RESSERVER;<Request ID="22" Action="CREATE"><Connection Source="IN_SERVER_SUBS.IND" Destination="OPCUASERVER_0.INIT" /></Request>
RESSERVER;<Request ID="23" Action="CREATE"><Connection Source="IN_SERVER_SUBS.IND" Destination="OPCUASERVER_1.INIT" /></Request>
RESSERVER;<Request ID="25" Action="CREATE"><Connection Source="OPCUASERVER_2.INITO" Destination="OUT_SERVER_PUBL.REQ" /></Request>
RESSERVER;<Request ID="26" Action="CREATE"><Connection Source="OPCUASERVER_2.QO" Destination="OUT_SERVER_PUBL.SD_1" /></Request>
RESSERVER;<Request ID="27" Action="CREATE"><Connection Source="OPCUASERVER_1.IND" Destination="LOG.REQ" /></Request>
RESSERVER;<Request ID="28" Action="CREATE"><Connection Source="OPCUASERVER_2.IND" Destination="LOG_0.REQ" /></Request>
RESSERVER;<Request ID="29" Action="CREATE"><Connection Source="IN_SERVER_SUBS.RD_1" Destination="OPCUASERVER_0.QI" /></Request>
RESSERVER;<Request ID="30" Action="CREATE"><Connection Source="IN_SERVER_SUBS.RD_1" Destination="OPCUASERVER_1.QI" /></Request>
RESSERVER;<Request ID="32" Action="CREATE"><Connection Source="OPCUASERVER_1.RD_1" Destination="LOG.VALUE" /></Request>
RESSERVER;<Request ID="33" Action="CREATE"><Connection Source="OPCUASERVER_2.RD_1" Destination="LOG_0.VALUE" /></Request>
RESSERVER;<Request ID="34" Action="START"/>

```

OPC UA and LOG FBs created

Connection between FBs

Figura 63: Archivo de configuración XML para crear servidores OPC UA

de la norma IEC 61499 (ver Figura 63), se puede ver en la primera parte del archivo las instrucciones para crear todos los SIFBs de los servidores y en la segunda parte las conexiones entre ellos y iii) Finalmente, con la opción SEND se descarga línea a línea al runtime IEC 61499 el archivo de configuración, cada vez que se envía una línea el runtime envía una respuesta la cual es presentada en la misma herramienta software. Estas variables estarán disponibles para ser descubiertas o servidas a clientes OPC-UA en otros pozos de manera fácil y segura.

4.9 CONCLUSIONES

En este capítulo se da un primer paso hacia la definición completa de una arquitectura basada en modelos para monitorizar el proceso de producción del petróleo que, utilizando estándares industriales consolidados (ISA 95 e ISA 88), puede ser la base para una integración vertical en el sector de la industria de producción de petróleo.

En la primera parte de este capítulo se ha especificado la información que necesita el servidor OPC-UA para dar respuesta a los requisitos de clientes remotos relativas a la monitorización del proceso de producción de petróleo. Como próximos pasos se contemplan, por un lado, la definición de los modelos restantes para completar la arquitectura (el PIM y el IEM), así como la generación automática del fichero de configuración del servidor OPC-UA y el diseño de la aplicación para la monitorización.

En la segunda parte del capítulo, se presenta un enfoque para acceder a los datos de campo en sistemas de automatización utilizando una herramienta de ingeniería basada en Java para crear servidores OPC-UA de manera sencilla bajo una arquitectura CPPS utilizando aplicaciones IEC 61499 que se ejecutan en runtimes de esta norma. El uso de este tipo de sistemas ayuda a introducir nuevos conceptos de CPPS dentro del paradigma de la Industria 4.0 en el proceso de Petróleo y Gas. Esta arquitectura proporciona una infraestructura M2M para las comunicaciones de la planta. Adicionalmente se propone un conjunto de SIFBs que se utilizan para implementar servidores OPC-UA y almacenar datos en bases de datos o software de tipo historian.

Este documento también describe un entorno de desarrollo integrado (IDE) creado de acuerdo con el metamodelo del Perfil de Cumplimiento IEC 61499 para Demostración de Factibilidad (CPFD), que permitirá a los ingenieros de petróleo y gas crear aplicaciones de control y comunicaciones industriales para integración vertical de datos a nivel de planta. La introducción de OPC-UA como un protocolo de comunicación general y universal permite incrementar la flexibilidad de la norma IEC 61499 y usarla en nuevos tipos de procesos. El método propuesto se basa en el modelo de información de OPC-UA, que permite no solo almacenar datos, sino también almacenarlos en una red estructurada de nodos interconectados.

El uso de este nuevo IDE para diseñar CPPS tiene muchos beneficios obvios y potenciales. Quizás el más simple pero el más importante es que el mismo archivo de configuración IEC 61499 es aplicable para diferentes procesos mediante reconfiguración de software. También es posible agregar, eliminar y reemplazar algunas funciones de un IC

en un servicio de comunicación cuando cambian los requisitos de la aplicación. Otra ventaja importante es la simplicidad del diseño y desarrollo, ya que es posible construir un sistema con componentes de software reutilizables.



5

CONCLUSIONES Y FUTURAS LÍNEAS DE INVESTIGACIÓN

El trabajo realizado da solución a los problemas planteados al inicio de esta tesis. Por otro lado, durante este estudio surgieron nuevas ideas, las cuales se presentan como futuras líneas de investigación dirigidas a afrontar los nuevos cambios en las aplicaciones de control distribuidas.

En este capítulo se resaltan las aportaciones originales como resultado de esta tesis. Asimismo, se proporcionan los comentarios que ayudan a entender el trabajo desarrollado y el alcance de la tesis. También se presentan las conclusiones que se destacan a partir de la elaboración del presente trabajo.

En un último punto, se presentan las futuras líneas de investigación que se plantean realizar a partir de esta tesis.

5.1 CONCLUSIONES Y PRINCIPALES APORTACIONES

Este trabajo se ha centrado en ofrecer un entorno adecuado para el desarrollo de aplicaciones distribuidas proporcionando de esta manera esquemas de comunicación M2M aplicadas al proceso de producción de petróleo y gas empleando el estándar IEC

61499, integrando de esta manera a esta industria a los conceptos dados por la Industria 4.0. IEC 61499 propone una metodología que utiliza el bloque de función SIFB, como el principal componente en la fase de diseño, siendo además independiente de la implementación.

Dentro de este trabajo de tesis, se han presentado varias aportaciones preliminares que han servido de base para la aportación final. La *aportación esencial* es la herramienta de software diseñada para permitir la *composición automática* de Componentes de Información distribuidas basados en IEC 61499 mediante la *inserción de SIFBs de servidores y clientes* implementadas mediante el uso del protocolo OPC-UA, para satisfacer los requisitos dados por la arquitectura CPPS flexible y acorde a las necesidades de la industria.

Es importante resaltar que el estándar IEC 61499 utiliza un modelo de ejecución basado en eventos y propone una arquitectura basada en modelos para representar cada uno de los elementos que conforman un sistema de control distribuido.

A lo largo del trabajo se han utilizado diferentes herramientas software para comprobar que cumplen con el estándar IEC 61499, identificando que sólo en algunas existe compatibilidad. Por otro lado, se identificó la falta de una metodología para el diseño y desarrollo de SIFBs de comunicaciones y control. Esto dio lugar a una de las aportaciones de la tesis, una metodología de diseño para crear SIFBs de acceso a hardware y SIFBs de comunicaciones cumpliendo con el estándar IEC 61499. Uno de los primeros puntos que motivaron la realización de esta metodología es que hoy en día no existe información documentada de cómo diseñar y programar los SIFBs y posteriormente utilizarlos para enlazar las aplicaciones con recursos de proceso y los sistemas de comunicación, así como la falta de información más específica de cómo desarrollarlos.

Por otra parte, el estándar solamente menciona la descripción general y las especificaciones de los SIFBs. Además, en la documentación referente al estándar IEC 61499 sólo se menciona la descripción general y el comportamiento de los SIFBs de comunicaciones proporcionados por la norma. El siguiente motivo fue, como se ha mencionado en

el capítulo 3 del estado del arte, que varios autores han realizado SIFBs específicos para una aplicación, pero no proporcionan información de cómo se realizaron.

Uno de los resultados obtenidos en esta investigación es la creación de plantillas SIFBs de servidores y clientes basados en el protocolo de comunicación OPC-UA. El conjunto de SIFBs desarrollados permite la integración vertical de los datos del proceso real a través de dispositivos entradas/salidas que usan distintos protocolos de comunicación como, por ejemplo, Modbus TCP o comunicaciones S7, y cumplen con una arquitectura CPPS anteriormente descrita.

Los resultados de esta aportación han dado lugar a las siguientes publicaciones:

1. **M. V. García**, F. Pérez, I. Calvo, F. López, and G. Morán, “Desarrollo de CPPS sobre IEC-61499 Basado en Dispositivos de Bajo Coste,” *XXXVI Jornadas de Automática*, Bilbao, Spain, pp. 230–237, 2015.
2. G. Morán, **M. V. García**, F. Pérez, “Control de una planta industrial utilizando sistemas de Bajo Coste”, *Ideas en Ciencia*, Num 44, Año 24, pp. 19-31, 2016.
3. **M. V. Garcia**, E. Irisarri, F. Perez, E. Estevez, D. Orive, and M. Marcos, “Integración de Comunicaciones a Nivel de Planta empleado Sistemas Ciber-físicos de Producción de Bajo Coste”, *XXXVII Jornadas de Automática*, Madrid, Spain, pp. 234–241, 2016.
4. E. Irisarri, **M. V. Garcia**, F. Perez, E. Estevez, D. Orive, and M. Marcos, “Propuesta Basada en Modelos para la Monitorización Remota de Procesos en la Industria de Producción de Petróleo”, *XXXVII Jornadas de Automática*, Madrid, Spain, pp. 331–337, 2016.
5. **M. V. Garcia**, F. Perez, I. Calvo, and G. Moran, “Developing CPPS within IEC-61499 based on low cost devices,” *Proceedings of IEEE International Workshop on Factory Communication Systems (WFCS'2015)*, Palma de Mallorca, Spain, vol. 2015–July, pp. 1–4, 2015.
6. **M. V. Garcia**, F. Perez, I. Calvo, and G. Moran, “Building industrial CPS with the IEC 61499 standard on low-cost hardware platforms,” *Proceedings of 2014*

- IEEE International Conference on Emerging Technologies and Factory Automation (ETFA2014)*., Luxembourg, Luxembourg, pp. 1–4, 2014.
7. **M. V. Garcia**, E. Irisarri, F. Perez, E. Estevez, D. Orive, and M. Marcos, “Plant floor communications integration using a low cost CPPS architecture,” *Proceedings of 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA2016)*, Berlin, Germany, pp. 1–4, 2016.
 8. E. Irisarri, **M. V. Garcia**, F. Perez, E. Estevez, and M. Marcos, “A model-based approach for process monitoring in oil production industry,” *Proceedings of 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA2016)*., Berlin, Germany, pp. 1–4, 2016.
 9. **M. V. Garcia**, E. Irisarri, F. Perez, E. Estevez, and M. Marcos, “OPC-UA communications integration using a CPPS architecture,” *Proceedings of 2016 IEEE Ecuador Technical. Chapters Meeting*, Guayaquil, Ecuador, pp. 1–6, 2016.

Por otro lado, también se identificó la falta de una metodología para desarrollar aplicaciones CPPS de control distribuidas integrando una arquitectura flexible, lo que permitió otra aportación al trabajo, una metodología de diseño y desarrollo de archivos XML de configuración para su uso en servidores OPC-UA permitiendo de esta manera la integración vertical de variables usadas a nivel de planta. Esta metodología está diseñada para que se pueda aplicar a cualquier herramienta de diseño de documentos XML, en consecuencia, es independiente de la herramienta usada. Cabe destacar que con esta propuesta será posible disminuir el tiempo de desarrollo de archivos de configuración para SIFBs de acceso a hardware y de comunicaciones basados en el protocolo OPC-UA y, en general, de la aplicación de automatización completa.

Finalmente, se ha diseñado y desarrollado la herramienta software de ingeniería cuyo principal objetivo es la introducción automática de los SIFBs de servidores OPC UA de comunicación necesarios para integración vertical de variables a nivel de planta y que puedan ser usados de manera sencilla y rápida en la industria del petróleo y gas, permitiendo que esta norma se integre en procesos reales.

Esta generación automática de la herramienta de ingeniería requiere de diferentes herramientas software. Una de las características de todas estas herramientas empleadas es que son de libre distribución. También se requiere de librerías generales, librerías propias y librerías de terceros.

Se ha realizado la verificación de la herramienta sobre un conjunto de aplicaciones distribuidas. Los resultados obtenidos al aplicar esta herramienta son satisfactorios. Es decir, la generación del nuevo sistema y creación de los componentes de información, da lugar a una aplicación completa y ejecutable en una de las herramientas hardware existentes.

Estos resultados han dado lugar a las siguientes publicaciones:

10. **M. V. Garcia**, E. Irisarri, F. Perez, M. Marcos, and E. Estevez, “Engineering tool to develop CPPS based on IEC-61499 and OPC UA for oil&gas process,” in *Proceedings of 2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS'2017)*, 2017, pp. 1–9.
11. **M. V. Garcia**, E. Irisarri, F. Perez, E. Estevez, and M. Marcos, “An Open CPPS Automation Architecture based on IEC-61499 over OPC-UA for flexible manufacturing in Oil&Gas Industry,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 1231–1238, Jul. 2017.
12. A. García, E. Castellanos, C. Rosero, C. Sánchez, **M. V. Garcia**, F. Perez and M. Marcos, “CPPS on Low Cost Devices for Batch Process under IEC-61499 and ISA-88,” *Proceedings of IEEE 15TH International Conference of Industrial Informatics (INDIN'2017)*, Emden, Germany, pp. 855-860, 2017.
13. S. Altamirano, R. Galleguillos, **M. V. Garcia**, F. Perez and M. Marcos, “Low Cost CPPs for Industrial Control under FHP algorithm,” *Proceedings of 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA2017)*, Limassol, Chipre, 2017.
14. **M. V. Garcia**, E. Irisarri, F. Perez, E. Estevez, and M. Marcos, “Arquitectura de Automatización basada en Sistemas Ciberfísicos para la Fabricación Flexible en

la Industria de Petróleo y Gas,” *Revista Iberoamericana de Automática e Informática Industrial RIAI*, vol 14, 2017.

Se obtienen beneficios tanto para el estándar como para el usuario, por ser independiente de la herramienta software. Esta herramienta es modular, fácil de extender, incluye automáticamente los SIFBs de comunicaciones basadas en servidores OPC UA, es independiente del software y herramientas basados en la norma IEC 61499 comerciales o de distribución gratuita y reduce el tiempo de desarrollo de aplicaciones de control distribuidas.

5.2 FUTURAS LÍNEAS DE INVESTIGACIÓN

La Tesis elaborada ha creado las bases de forma conceptual y de desarrollo que han permitido encontrar una serie de futuras líneas de investigación de gran interés para continuar avanzando sobre este tema resaltando las siguientes:

1. Los resultados obtenidos demuestran que es viable continuar investigando sobre este tema, buscando nuevos objetivos adaptándose a las nuevas tendencias tecnológicas y a su uso en procesos industriales reales y de alta complejidad de automatización.
2. Se hace necesaria la ampliación de la librería de SIFBs para dotar al diseñador de sistemas de control distribuidos de un conjunto de redes suficiente como para poder resolver cualquier tipo de comunicación industrial, incluyendo protocolos como MQTT o AMQP.
3. Las metodologías y herramientas desarrolladas en esta tesis abren el campo para la posible integración de nuevas herramientas y funcionalidades compatibles con el estándar IEC 61499.
4. Se ha visto pertinente que se debe implementar servicios como Informes (*Report*), Consultar (*Query*), implementación de Niveles de Seguridad (*Security*), Almacenar datos históricos del proceso (*Historian*) para cumplir con todos los objetivos de los Componentes de Información (IC) propuestos en la Arquitectura CPPS.

5. Se puede considerar la implementación de un runtime IEC 61499 como FORTE sobre servidores OPC UA, de esta manera se lograría un servidor genérico que puede solucionar diversos problemas en la automatización actual y no solo enfocarse a aplicaciones IEC 61499.
6. Dentro del modelado de los sistemas y aplicaciones industriales, podría realizarse una extensión del trabajo desarrollado hacia su integración dentro de un formato dado por AutomationML que permita la interoperabilidad entre herramientas de ingeniería.

Bibliografía

- [1] A. El Zawawi and A. El-Sayed, “Integration of DCS and ESD through an OPC application for upstream Oil and Gas,” *IEEE Power Energy Soc. Gen. Meet.*, pp. 1–5, 2012.
- [2] O. Sande, M. Fojcik, and R. Cupek, “OPC UA Based Solutions for Integrated Operations,” *Commun. Comput. Inf. Sci.*, vol. 79, pp. 76–83, 2010.
- [3] M. Floeck, A. Papageorgiou, A. Schuelke, and J. Song, “Horizontal M2M platforms boost vertical industry: Effectiveness study for building energy management systems,” *2014 IEEE World Forum Internet Things, WF-IoT 2014*, pp. 15–20, 2014.
- [4] International Electrotechnical Commission, “International Electrotechnical Commission Std. (2005) IEC 61499: Function blocks, Part 1-4,” 2014. [Online]. Available: <http://www.iec.ch>.
- [5] K. Windt, F. Böse, and T. Philipp, “Autonomy in production logistics: Identification, characterisation and application,” *Robot. Comput. Integr. Manuf.*, vol. 24, no. 4, pp. 572–578, 2008.
- [6] G. Neugschwandtner, M. Reekmans, and D. van der Linden, “An open automation architecture for flexible manufacturing,” in *2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA)*, 2013, pp. 1–5.
- [7] J. Kim, J. Lee, J. Kim, and J. Yun, “M2M service platforms: Survey, issues, and enabling technologies,” *IEEE Commun. Surv. Tutorials*, vol. 16, no. 1, pp. 61–76, 2014.
- [8] N. Jazdi, “Cyber physical systems in the context of Industry 4.0,” *Autom. Qual. Testing, Robot. 2014 IEEE ...*, pp. 2–4, 2014.
- [9] C. R. Baudoin and K. Management, “Deploying the Industrial Internet in Oil & Gas: Challenges and Opportunities,” *Soc. Pet. Eng.*, no. September, pp. 6–8, 2016.
- [10] C. M. Cheung, P. Goyal, G. Harris, O. Patri, A. Srivastava, Y. Zhang, A. Panangadan, C. Chelmiss, R. McKee, M. Theron, T. Nemeth, and V. K. Prasanna, “Rapid Data Integration and Analysis for Upstream Oil and Gas Applications,” in *SPE Annual Technical Conference and Exhibition*, 2015.
- [11] A. Ambrus, P. Ashok, and E. van Oort, “Drilling Rig Sensor Data Validation in the Presence of Real-Time Process Variations,” in *SPE Annual Technical Conference and Exhibition*, 2013, no.

SEPTEMBER 2013.

- [12] F. Perez, E. Irisarri, D. Orive, M. Marcos, and E. Estevez, "A CPPS Architecture approach for Industry 4.0," in *2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFa)*, 2015, pp. 1–4.
- [13] A. Mazak and C. Huemer, "HoVer: A modeling framework for horizontal and vertical integration," in *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)*, 2015, pp. 1642–1647.
- [14] L. Wang and G. Wang, "Big Data in Cyber-Physical Systems, Digital Manufacturing and Industry 4.0," *I.J. Eng. Manuf.*, vol. 4, no. July, pp. 1–8, 2016.
- [15] J. Wan, S. Tang, Z. Shu, D. Li, S. Wang, M. Imran, and A. Vasilakos, "Software-Defined Industrial Internet of Things in the Context of Industry 4.0," *IEEE Sens. J.*, pp. 1–1, 2016.
- [16] G.-J. Cheng, L.-T. Liu, X.-J. Qiang, and Y. Liu, "Industry 4.0 Development and Application of Intelligent Manufacturing," in *2016 International Conference on Information System and Artificial Intelligence (ISAI)*, 2016, pp. 407–410.
- [17] H. Cheng, P. Zeng, L. Xue, Z. Shi, P. Wang, and H. Yu, "Manufacturing Ontology Development Based on Industry 4.0 Demonstration Production Line," in *2016 Third International Conference on Trustworthy Systems and their Applications (TSA)*, 2016, pp. 42–47.
- [18] K. Dequeant, P. Vialletelle, P. Lemaire, and M.-L. Espinouse, "A literature review on variability in semiconductor manufacturing: The next forward leap to Industry 4.0," in *2016 Winter Simulation Conference (WSC)*, 2016, pp. 2598–2609.
- [19] S. Shamim, S. Cang, H. Yu, and Y. Li, "Management approaches for Industry 4.0: A human resource management perspective," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, 2016, pp. 5309–5316.
- [20] M. Huba and S. Kozak, "From e-Learning to Industry 4.0," in *2016 International Conference on Emerging eLearning Technologies and Applications (ICETA)*, 2016, pp. 103–108.
- [21] H. Flatt, S. Schriegel, J. Jasperneite, H. Trsek, and H. Adamczyk, "Analysis of the Cyber-Security of industry 4.0 technologies based on RAMI 4.0 and identification of requirements," in *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFa)*, 2016, pp. 1–4.
- [22] E. M. Frazzon, J. Hartmann, T. Makuschewitz, and B. Scholz-Reiter, "Towards socio-cyber-physical systems in production networks," *Procedia CIRP*, vol. 7, pp. 49–54, 2013.

- [23] Y. Duan, W. Li, Y. Zhong, and X. Fu, "A multi-network control framework based on industrial internet of things," in *2016 IEEE 13th International Conference on Networking, Sensing, and Control (ICNSC)*, 2016, pp. 1–5.
- [24] F. Mattern and C. Floerkemeier, "From the internet of computers to the internet of things," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6462 LNCS, pp. 242–259, 2010.
- [25] Hao Wang, O. L. Osen, Guoyuan Li, Wei Li, Hong-Ning Dai, and Wei Zeng, "Big data and industrial Internet of Things for the maritime industry in Northwestern Norway," in *TENCON 2015 - 2015 IEEE Region 10 Conference*, 2015, pp. 1–5.
- [26] A. Ismail and W. Kastner, "A middleware architecture for vertical integration," in *2016 1st International Workshop on Cyber-Physical Production Systems (CPPS)*, 2016, pp. 1–4.
- [27] Z. Meng, Z. Wu, C. Muvianto, and J. Gray, "A Data-Oriented M2M Messaging Mechanism for Industrial IoT Applications," *IEEE Internet Things J.*, pp. 1–1, 2017.
- [28] R. Petrasch and R. Hentschke, "Process modeling for industry 4.0 applications: Towards an industry 4.0 process modeling language and method," in *2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, 2016, pp. 1–5.
- [29] M. V. Garcia, E. Irisarri, F. Perez, E. Estevez, and M. Marcos, "OPC-UA communications integration using a CPPS architecture," *2016 IEEE Ecuador Tech. Chapters Meet.*, pp. 1–6, 2016.
- [30] H. Sasajima, T. Ishikuma, and H. Hayashi, "Future IIOT in process automation — Latest trends of standardization in industrial automation, IEC/TC65," in *2015 54th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, 2015, pp. 963–967.
- [31] E. Darmois and O. Elloumi, "Introduction to M2M," in *M2M Communications*, Chichester, UK: John Wiley & Sons, Ltd, 2012, pp. 1–20.
- [32] Guo Wang, Xiaofeng Zhong, Shunliang Mei, and Jing Wang, "An adaptive medium access control mechanism for cellular based Machine to Machine (M2M) communication," in *2010 IEEE International Conference on Wireless Information Technology and Systems*, 2010, pp. 1–4.
- [33] E. A. Lee, "Cyber physical systems: Design challenges," *Object Oriented Real-Time Distrib. Comput. (ISORC), 11th IEEE Int. Symp.*, pp. 363–369, 2008.
- [34] K. Beetz, "Die Wirtschaftliche Bedeutung von Cyberphysical Systems aus der Sicht Eines Global Players," in *Cyber-Physical Systems: Innovation Durch Software-Intensive Eingebettete Systeme*, M. Broy, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 59–66.

- [35] Z. Zhang, J. Porter, E. Eyisi, G. Karsai, X. Koutsoukos, and J. Sztipanovits, “Co-simulation framework for design of time-triggered cyber physical systems,” in *Proceedings of the ACM/IEEE 4th International Conference on Cyber-Physical Systems - ICCPS '13*, 2013, p. 119.
- [36] McKinsey & Company, “Big data: The next frontier for innovation, competition, and productivity,” *McKinsey Glob. Inst.*, no. June, p. 156, 2011.
- [37] H. A. Simon, *The Sciences of the Artificial (3rd Ed.)*. Cambridge, MA, USA: MIT Press, 1996.
- [38] J. Lee, H. D. Ardakani, S. Yang, and B. Bagheri, “Industrial Big Data Analytics and Cyber-physical Systems for Future Maintenance & Service Innovation,” *Procedia CIRP*, vol. 38, pp. 3–7, 2015.
- [39] H.-A. Kao, W. Jin, D. Siegel, and J. Lee, “A Cyber Physical Interface for Automation Systems—Methodology and Examples,” *Machines*, vol. 3, no. 2, pp. 93–106, May 2015.
- [40] M. Broy, “Cyber-Physical Systems --- Wissenschaftliche Herausforderungen Bei Der Entwicklung,” in *Cyber-Physical Systems: Innovation Durch Software-Intensive Eingebettete Systeme*, M. Broy, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 17–31.
- [41] M. James, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. H. Byers, “Outline Cyber Physical Production systems / Industry 4 . 0- challenges in research and industrial application) – Production changes due to malfunctions – New products – Maintenance support Characteristics of Cyber-Physical Production Systems,” *Big data next Front. Innov. Compet. Product. June 2011*, pp. 1–20, 2015.
- [42] Kyoung-Dae Kim and P. R. Kumar, “Cyber–Physical Systems: A Perspective at the Centennial,” *Proc. IEEE*, vol. 100, no. Special Centennial Issue, pp. 1287–1308, May 2012.
- [43] L. Monostori, “Cyber-physical production systems: Roots, expectations and R&D challenges,” *Procedia CIRP*, vol. 17, pp. 9–13, 2014.
- [44] A. Canedo, E. Schwarzenbach, and M. A. Al Faruque, “Context-sensitive synthesis of executable functional models of cyber-physical systems,” in *Proceedings of the ACM/IEEE 4th International Conference on Cyber-Physical Systems - ICCPS '13*, 2013, p. 99.
- [45] F. Jammes and H. Smit, “Service-Oriented Paradigms in Industrial Automation,” *IEEE Trans. Ind. Informatics*, vol. 1, no. 1, pp. 62–70, Feb. 2005.
- [46] K. C. Thramboulidis, G. Doukas, and G. Koumoutsos, “A SOA-Based Embedded Systems Development Environment for Industrial Automation,” *EURASIP J. Embed. Syst.*, vol. 2008, no. 1, p. 312671, 2007.
- [47] R. U. I. Rodrigues and M. Disserta, “DPWS Middleware to support Agent-based Manufacturing

- Control and Simulation,” Universidade Nova de Lisboa, 2009.
- [48] W. Shen, Q. Hao, H. J. Yoon, and D. H. Norrie, “Applications of agent-based systems in intelligent manufacturing: An updated review,” *Adv. Eng. Informatics*, vol. 20, no. 4, pp. 415–431, Oct. 2006.
- [49] D. K. Nguyen and D. Savio, “Exploiting SOA for adaptive and distributed manufacturing with cross enterprise shop floor commerce,” in *Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services - iiWAS '08*, 2008, p. 318.
- [50] L. Richardson and S. Ruby, *RESTful Web Services*. 2007.
- [51] T. Dörnemann, M. Mathes, R. Schwarzkopf, E. Juhnke, and B. Freisleben, “DAVO: A Domain-Adaptable, Visual BPEL4WS Orchestrator,” in *2009 International Conference on Advanced Information Networking and Applications*, 2009, pp. 121–128.
- [52] U. Thakar, A. Tiwari, and S. Varma, “On Composition of SOAP Based and RESTful Services,” in *2016 IEEE 6th International Conference on Advanced Computing (IACC)*, 2016, pp. 500–505.
- [53] S. Kumari and S. K. Rath, “Performance comparison of SOAP and REST based Web Services for Enterprise Application Integration,” in *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2015, pp. 1656–1660.
- [54] R. T. Fielding, “Architectural Styles and the Design of Network-based Software Architectures,” University of California, Irvine, 2000.
- [55] P. Hazarika, S. Shenoy, S. B. Tolety, and N. Kalekar, “Mobile cloud integration for industrial data interchange,” *2015 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2015*, pp. 1118–1122, 2015.
- [56] S. Kozar and P. Kadera, “Integration of IEC 61499 with OPC UA,” in *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2016, pp. 1–7.
- [57] A. Claassen, S. Rohjans, and S. Lehnhoff Member, “Application of the OPC UA for the Smart Grid,” in *2011 2nd IEEE PES International Conference and Exhibition on Innovative Smart Grid Technologies*, 2011, pp. 1–8.
- [58] D. van der Linden, H. Mannaert, W. Kastner, V. Vanderputten, H. Peremans, and J. Verelst, “An OPC UA interface for an evolvable ISA88 control module,” in *ETFA2011*, 2011, pp. 1–9.
- [59] S. Sucic, B. Bony, L. Guise, F. Jammes, and A. Marusic, “Integrating DPWS and OPC UA device-level SOA features into IEC 61850 applications,” in *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, 2012, pp. 5773–5778.
- [60] M. Melik-Merkumians, T. Baier, M. Steinegger, W. Lepuschitz, I. Hegny, and A. Zoitl, “Towards

- OPC UA as portable SOA middleware between control software and external added value applications,” in *Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies & Factory Automation (ETFA 2012)*, 2012, pp. 1–8.
- [61] S. Rohjans, M. Uslar, and H. Juergen Appelrath, “OPC UA and CIM: Semantics for the smart grid,” in *IEEE PES T&D 2010*, 2010, pp. 1–8.
- [62] O. P. C. Unified, A. Specification, and C. Release, “OPC UA Foundation Part 1,” 2009, p. 28.
- [63] S. Lehnhoff, W. Mahnke, S. Rohjans, and M. Uslar, “IEC 61850 based OPC UA Communication - The Future of Smart Grid Automation,” *Power*, pp. 303–308, 2011.
- [64] A. Zoitl, C. Sunder, and I. Terzic, “Dynamic reconfiguration of distributed control applications with reconfiguration services based on IEC 61499,” in *IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS'06)*, 2006, pp. 109–114.
- [65] V. Vyatkin and J. Chouinard, “On comparisons of the ISaGRAF implementation of IEC 61499 with FBDK and other implementations,” in *2008 6th IEEE International Conference on Industrial Informatics*, 2008, pp. 289–294.
- [66] Wenbin Dai and V. Vyatkin, “Redesign distributed IEC 61131-3 PLC system in IEC 61499 function blocks,” in *2010 IEEE 15th Conference on Emerging Technologies & Factory Automation (ETFA 2010)*, 2010, pp. 1–8.
- [67] A. Zoitl and V. Vyatkin, “Different perspectives [Face to face; ‘IEC 61499 architecture for distributed automation: The ‘glass half full’ view,” *IEEE Ind. Electron. Mag.*, vol. 3, no. 4, pp. 7–23, 2009.
- [68] L. H. Yoong, P. S. Roop, V. Vyatkin, and Z. Salcic, “A Synchronous Approach for IEC 61499 Function Block Implementation,” *IEEE Trans. Comput.*, vol. 58, no. 12, pp. 1599–1614, Dec. 2009.
- [69] H. Kaghazchi, R. Joyce, and D. Heffernan, “A function block diagnostic framework for a multi-vendor PROFIBUS environment,” *Assem. Autom.*, vol. 27, no. 3, pp. 240–246, 2007.
- [70] K. Thramboulidis, “IEC 61499 function block model: Facts and fallacies,” *IEEE Ind. Electron. Mag.*, vol. 3, no. 4, pp. 7–23, 2009.
- [71] K. Thramboulidis, “A Model Based Approach to Address Inefficiencies of the IEC61499 Function Block Model,” no. Icssea, pp. 1–9, 2006.
- [72] Holobloc Inc., “FBDK 2.1 - The Function Block Development Kit.” [Online]. Available: <http://www.holobloc.com/fbdk2/>.
- [73] 4DIAC Consortium. PROFACTOR GmbH., “No Title,” *Framework for Distributed Industrial*

Automation and Control (4DIAC), 2010. [Online]. Available: <http://www.fordiac.org>.

- [74] A. Zoitl, T. Strasser, and G. Ebenhofer, "Developing modular reusable IEC 61499 control applications with 4DIAC," *IEEE Int. Conf. Ind. Informatics*, pp. 358–363, 2013.
- [75] V. Vyatkin and I. S. of America, *IEC 61499 function blocks for embedded and distributed control systems design*. ISA-Instrumentation, Systems, and Automation Society, 2007.
- [76] L. Wang, R. W. Brennan, S. Balasubramanian, and D. H. Norrie, "Realizing Holonic Control with Function Blocks," *Integr. Comput. Eng.*, vol. 8, no. 1, pp. 81–93, 2001.
- [77] C. Schwab, M. Tangermann, and L. Ferrarini, "Web based methodology for engineering and maintenance of distributed control systems: the TORERO approach," in *INDIN '05. 2005 3rd IEEE International Conference on Industrial Informatics, 2005.*, 2005, pp. 32–37.
- [78] N. Völker and B. J. Krämer, "Automated verification of function block based industrial control systems," *Electron. Notes Theor. Comput. Sci.*, vol. 25, no. July 1999, pp. 97–110, 1999.
- [79] K. Thramboulidis and C. Tranoris, "An architecture for the development of function block oriented engineering support systems," in *Proceedings 2001 IEEE International Symposium on Computational Intelligence in Robotics and Automation (Cat. No.01EX515)*, 2001, pp. 536–542.
- [80] R. W. Brennan, X. Zhang, Y. Xu, and D. H. Norrie, "A Reconfigurable Concurrent Function Block Model and Its Implementation in Real-time Java," *Integr. Comput. Eng.*, vol. 9, no. 3, pp. 263–279, 2002.
- [81] S. Olsen, J. Wang, A. Ramirez-Serrano, and R. W. Brennan, "Contingencies-based reconfiguration of distributed factory automation," *Robot. Comput. Integr. Manuf.*, vol. 21, no. 4–5, pp. 379–390, 2005.
- [82] A. Zoitl, G. Grabmair, F. Auinger, and C. Sunder, "Executing real-time constrained control applications modelled in IEC 61499 with respect to dynamic reconfiguration," in *INDIN '05. 2005 3rd IEEE International Conference on Industrial Informatics, 2005.*, 2005, pp. 62–67.
- [83] T. Strasser, A. Zoitl, J. H. Christensen, and C. Sünder, "Design and Execution Issues in IEC 61499 Distributed Automation and Control Systems," *IEEE Trans. Syst. Man, Cybern. Part C (Applications Rev.)*, vol. 41, no. 1, pp. 41–51, Jan. 2011.
- [84] T. Strasser, F. Auinger, and A. Zoitl, "Development, implementation and use of an IEC 61499 function block library for embedded closed loop control," in *2nd IEEE International Conference on Industrial Informatics, 2004. INDIN '04. 2004*, 2004, pp. 594–599.
- [85] S. Jain, C. Yuan, and P. Ferreira, "EMBench: A Rapid Prototyping Environment for Numerical

- Control Systems,” in *Dynamic Systems and Control*, 2002, vol. 2002, pp. 7–13.
- [86] C. Yuan and P. Ferreira, “An Integrated Environment for the Design and Control of Deadlock-Free Flexible Manufacturing Cells,” in *Manufacturing Engineering and Materials Handling Engineering*, 2004, vol. 2004, pp. 471–481.
- [87] C. Yuan and P. Ferreira, “An Integrated Rapid Prototyping Environment for Reconfigurable Manufacturing Systems,” in *Materials*, 2003, vol. 2003, pp. 737–744.
- [88] T. Hussain and G. Frey, “Developing IEC 61499 compliant distributed systems with network enabled controllers,” in *IEEE Conference on Robotics, Automation and Mechatronics, 2004.*, 2004, vol. 1, pp. 507–512.
- [89] L. Wang, S. Keshavarzmanesh, and H. Y. Feng, “Design of adaptive function blocks for dynamic assembly planning and control,” *J. Manuf. Syst.*, vol. 27, no. 1, pp. 45–51, 2008.
- [90] L. Wang, Y. Song, and Q. Gao, “Designing function blocks for distributed process planning and adaptive control,” *Eng. Appl. Artif. Intell.*, vol. 22, no. 7, pp. 1127–1138, 2009.
- [91] P. A. Vicaire, E. Hoque, S. Member, Z. Xie, and J. A. Stankovic, “Bundle : A Group-Based Programming Abstraction for Cyber-Physical Systems,” vol. 8, no. 2, pp. 379–392, 2012.
- [92] I. Stojmenovic, “Machine-to-Machine Communications with In-network Data Aggregation, Processing and Actuation for Large Scale Cyber-Physical Systems,” *IEEE Internet Things J.*, vol. PP, no. 99, pp. 1–1, 2014.
- [93] W. Kang, K. Kapitanova, and S. Son, “RDDS: A real-time data distribution service for cyber-physical systems,” *IEEE Trans. Ind. Informatics*, vol. 8, no. 2, pp. 393–405, 2012.
- [94] J. Lee, B. Bagheri, and H.-A. Kao, “Recent Advances and Trends of Cyber-Physical Systems and Big Data Analytics in Industrial Informatics,” *Int. Conf. Ind. Informatics 2014*, no. November 2015, 2014.
- [95] J. Lee, E. Lapira, B. Bagheri, and H. an Kao, “Recent advances and trends in predictive manufacturing systems in big data environment,” *Manuf. Lett.*, vol. 1, no. 1, pp. 38–41, 2013.
- [96] W. Dai and C. Chen, “Modeling Distributed Automation Systems in Cyber- Physical View,” pp. 984–989, 2015.
- [97] L. Ferrarini, C. Veber, and P. Milano, “Design and implementation of distributed hierarchical automation and control systems with IEC 61499,” *I Can*, pp. 0–5, 2005.
- [98] L. Ferrarini, C. Veber, and G. Fogliazza, “IEC 61499 implementation of a Modular Control Model for Manufacturing Systems,” in *2005 IEEE Conference on Emerging Technologies and Factory*

Automation, vol. 1, pp. 315–321.

- [99] K. Thramboulidis, “Model-Integrated Mechatronics—Toward a New Paradigm in the Development of Manufacturing Systems,” *IEEE Trans. Ind. Informatics*, vol. 1, no. 1, pp. 54–61, Feb. 2005.
- [100] J.H.Christensen, “Design patterns for systems engineering in IEC 61499,” in *Verteilte Automatisierung - Modelle und Methoden für Entwurf, Verifikation, Engineering und Instrumentierung*, Germany, 2000, pp. 63–71.
- [101] M. Rentschler, H. Trsek, and L. Durkop, “OPC UA extension for IP auto-configuration in cyber-physical systems,” in *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*, 2016, pp. 26–31.
- [102] W. Dai, V. Vyatkin, J. H. Christensen, and V. N. Dubinin, “Bridging Service-Oriented Architecture and IEC 61499 for Flexibility and Interoperability,” *Ind. Informatics, IEEE Trans.*, vol. 11, no. 3, pp. 771–781, 2015.
- [103] K. Thramboulidis, “IEC 61499 vs. 61131: A Comparison Based on Misperceptions,” *J. Softw. Eng. Appl.*, vol. 6, no. August, pp. 405–415, 2013.
- [104] J. Parra, M. A. Hossain, A. Uribarren, E. Jacob, and A. El Saddik, “Flexible smart home architecture using device profile for web services: A peer-to-peer approach.”
- [105] P. E. Rovsing, P. G. Larsen, and T. S. Toftegaard, “A Reality Check on Home Automation Technologies,” *J. Green Eng.*, vol. 1, no. 3, pp. 303–327, 2011.
- [106] C. Schwab, M. Tangermann, and A. Lueder, “The Modular TORERO IEC 61499 Engineering Platform - Eclipse in Automation,” in *2005 IEEE Conference on Emerging Technologies and Factory Automation*, vol. 2, pp. 265–272.
- [107] F. Andren and T. Strasser, “Distributed open source control with Industrial Ethernet I/O devices,” in *ETFA2011*, 2011, pp. 1–4.
- [108] J. J. Scarlett and R. W. Brennan, “Evaluating a new communication protocol for real-time distributed control,” *Robot. Comput. Integr. Manuf.*, vol. 27, no. 3, pp. 627–635, Jun. 2011.
- [109] F. Andren, R. Brundlinger, and T. Strasser, “IEC 61850/61499 Control of Distributed Energy Resources: Concept, Guidelines, and Implementation,” *IEEE Trans. Energy Convers.*, vol. 29, no. 4, pp. 1008–1017, Dec. 2014.
- [110] N. Higgins, V. Vyatkin, N.-K. C. Nair, and K. Schwarz, “Distributed Power System Automation With IEC 61850, IEC 61499, and Intelligent Control,” *IEEE Trans. Syst. Man, Cybern. Part C*

(*Applications Rev.*, vol. 41, no. 1, pp. 81–92, Jan. 2011.

- [111] M. Erol-Kantarci and H. T. Mouftah, “Wireless multimedia sensor and actor networks for the next generation power grid,” *Ad Hoc Networks*, vol. 9, no. 4, pp. 542–551, 2011.
- [112] H. (Deutsche P. A. Henning, Kagermann(National Academy of Science and Engineering). Wolfgang, Wahlster (German Research Center for Artificial Intelligence). Johannes, “Recommendations for implementing the strategic initiative INDUSTRIE 4.0,” *Final Rep. Ind. 4.0 WG*, no. April, p. 82, 2013.
- [113] K. J. Aström, “Event Based Control,” in *Analysis and Design of Nonlinear Control Systems*, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 127–147.
- [114] K.-E. Årzén, “A simple event-based PID controller,” *IFAC Proc. Vol.*, vol. 32, no. 2, pp. 8687–8692, Jul. 1999.
- [115] J. Yan and V. Vyatkin, “Distributed Software Architecture Enabling Peer-to-Peer Communicating Controllers,” *IEEE Trans. Ind. Informatics*, vol. 9, no. 4, pp. 2200–2209, Nov. 2013.
- [116] M. Sorouri, S. Patil, and V. Vyatkin, “Distributed control patterns for intelligent mechatronic systems,” in *IEEE 10th International Conference on Industrial Informatics*, 2012, pp. 259–264.
- [117] D. Baigent, M. Adamiak, R. Mackiewicz, and G. E. M. G. E. M. SISCO, “IEC 61850 communication networks and systems in substations: An overview for users.”
- [118] J. Lee, S.-J. Lee, and P.-F. Wang, “A Framework for Composing SOAP, Non-SOAP and Non-Web Services,” *IEEE Trans. Serv. Comput.*, vol. 8, no. 2, pp. 240–250, Mar. 2015.
- [119] J. Yan, C. Pang, C.-W. Yang, and V. Vyatkin, “Adaptable software components: Towards digital ecosystems and software evolution in the industrial automation domain,” in *IECON 2014 - 40th Annual Conference of the IEEE Industrial Electronics Society*, 2014, pp. 2512–2518.
- [120] M. V. Garcia, E. Irisarri, F. Perez, M. Marcos, and E. Estevez, “Engineering tool to develop CPPS based on IEC-61499 and OPC UA for oil&gas process,” *2017 IEEE 13th Int. Work. Fact. Commun. Syst.*, pp. 1–9, May 2017.
- [121] M. V. García, F. Pérez, I. Calvo, and G. Morán, “Building industrial CPS with the IEC 61499 standard on low-cost hardware platforms,” *19th IEEE Int. Conf. Emerg. Technol. Fact. Autom. ETFA 2014*, 2015.
- [122] M. V. Garcia, F. Perez, I. Calvo, and G. Moran, “Developing CPPS within IEC-61499 based on low cost devices,” *IEEE Int. Work. Fact. Commun. Syst. - Proceedings, WFCS*, vol. 2015–July, pp. 1–4, 2015.

- [123] R. Henßen and M. Schleipen, “Interoperability between OPC UA and AutomationML,” *Procedia CIRP*, vol. 25, pp. 297–304, 2014.
- [124] IEC, “IEC 62769-1 CDV: Field Device Integration (FDI), Technical Specification - Part 1: Overview.,” 2013.
- [125] M. Compton, P. Barnaghi, L. Bermudez, R. García-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, V. Huang, K. Janowicz, W. D. Kelsey, D. Le Phuoc, L. Lefort, M. Leggieri, H. Neuhaus, A. Nikolov, K. Page, A. Passant, A. Sheth, and K. Taylor, “The SSN ontology of the W3C semantic sensor network incubator group,” *Web Semant. Sci. Serv. Agents World Wide Web*, vol. 17, pp. 25–32, Dec. 2012.
- [126] F. Pérez González, E. Estevez, I. Calvo, D. Orive, and M. Marcos, *Fundamentals of the IEC 61400-25 Standard: Communications for Monitoring and Control of Wind Power Plants*. 2014.
- [127] W. Mahnke, S.-H. Leitner, and M. Damm, *OPC Unified Architecture*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.
- [128] ISA, “ISA95, Enterprise-Control System Integration,” 2017. [Online]. Available: <https://www.isa.org/isa95/>.
- [129] ISA, *ANSI/ISA-88.01 Batch Control Part 1 : Models and Terminology*, vol. 1, no. October. 1995.
- [130] K. Nagorny, R. Harrison, A. W. Colombo, and G. Kreutz, “A formal engineering approach for control and monitoring systems in a service-oriented environment,” in *2013 11th IEEE International Conference on Industrial Informatics (INDIN)*, 2013, pp. 480–487.
- [131] D. Dimitrova, S. Panjaitan, I. Batchkova, and G. Frey, “IEC 61499 Component Based Approach for Batch Control Systems,” *IFAC Proc. Vol.*, vol. 41, no. 2, pp. 10875–10880, 2008.
- [132] S. M. Kannan, K. Suri, J. Cadavid, I. Barosan, M. van den Brand, M. Alferez, and S. Gerard, “Towards Industry 4.0: Gap Analysis between Current Automotive MES and Industry Standards Using Model-Based Requirement Engineering,” in *2017 IEEE International Conference on Software Architecture Workshops (ICSAW)*, 2017, pp. 29–35.
- [133] D. Ivanova, I. Batchkova, S. Panjaitan, F. Wagner, and G. Frey, “Combining IEC 61499 and ISA S88 for Batch Control,” *IFAC Proc. Vol.*, vol. 42, no. 4, pp. 187–192, 2009.
- [134] K. H. Hall, R. J. Staron, and A. Zoitl, “Challenges to Industry Adoption of the IEC 61499 Standard on Event-based Function Blocks,” in *2007 5th IEEE International Conference on Industrial Informatics*, 2007, pp. 823–828.
- [135] C. C. Insaurralde, “Modeling standard for distributed control systems: IEC 61499 from industrial

- automation to aerospace,” in *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, 2016, pp. 1–8.
- [136] D. Drozdov, V. Dubinin, and V. Vyatkin, “Speculative computation in IEC 61499 function blocks execution — Modeling and simulation,” in *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*, 2016, pp. 748–755.
- [137] D. Brandl, *Design Patterns for Flexible Manufacturing*. ISA, 2006.
- [138] S. Liu, H. Shi, and Z. Yang, “Research of Control Recipe for Batch Plants Based on PFC,” in *2010 International Conference on Measuring Technology and Mechatronics Automation*, 2010, pp. 1079–1082.
- [139] Zhihua Xiong, Jie Zhang, Xiong Wang, and Yongmao Xu, “An integrated batch-to-batch iterative learning control and within batch control strategy for batch processes,” in *Proceedings of the 2005, American Control Conference, 2005.*, pp. 1935–1940.
- [140] A. McDonald and M. P. Givens, “Repeated S88 Success Yields Cost Reductions at Large Consumer Products Company,” no. April, pp. 1–10, 2001.
- [141] M. Melik Merikumians, M. Baierling, and G. Schitter, “A service-oriented domain specific language programming approach for batch processes,” in *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2016, pp. 1–9.
- [142] M. De Minicis, F. Giordano, F. Poli, and M. M. Schiraldi, “Recipe Development Process Re-Design with ANSI/ISA-88 Batch Control Standard in the Pharmaceutical Industry,” *Int. J. Eng. Bus. Manag.*, vol. 6, p. 16, Jan. 2014.
- [143] F. Perez, E. Irisarri, D. D. Orive, M. Marcos, E. Estevez, F. Perez, E. Irisarri, D. D. Orive, M. Marcos, and E. Estevez, “A CPPS Architecture approach for Industry 4.0,” in *2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)*, 2015, vol. 2015–Octob, pp. 1–4.
- [144] D. HE, “An Approach for ISA-95 Application to Industrial Systems,” Tempere University of Technology.
- [145] Muhammad Younus, Cong Peiyong, Lu Hu, and Fan Yuqing, “MES development and significant applications in manufacturing -A review,” in *2010 2nd International Conference on Education Technology and Computer*, 2010, pp. V5-97-V5-101.
- [146] M. de Sousa, “Analyzing the compatibility between ISA 88 and IEC 61499,” in *2010 IEEE 15th Conference on Emerging Technologies & Factory Automation (ETFA 2010)*, 2010, pp. 1–8.

- [147] G. W. Hood, "Industry standard models for design/discussion of manufacturing systems," *2015 5th Aust. Control Conf.*, pp. 12–15, 2015.
- [148] S. Choi, K. Jung, B. Kulvatunyou, and K. Morris, "An analysis of technologies and standards for designing smart manufacturing systems," *J. Res. Natl. Inst. Stand. Technol.*, vol. 121, p. 422, Sep. 2016.
- [149] Isagraf, "IEC 61131 and IEC 61499 Software." [Online]. Available: <http://www.isagraf.com/index.htm>.
- [150] ntxControl, "ntxControl." [Online]. Available: <http://www.nxtcontrol.com/>.
- [151] 4DIAC, "IEC 61499 Implementation for Distributed," 2017. [Online]. Available: <https://eclipse.org/4diac/>.
- [152] Elsisit, "NETMASTER." [Online]. Available: <http://www.elsist.it>.
- [153] Tait Controls, "TaitControls." [Online]. Available: <http://www.taitcontrols.co.nz/>.
- [154] K. C. Thramboulidis and C. S. Tranoris, "Developing a CASE tool for distributed control applications," *Int. J. Adv. Manuf. Technol.*, vol. 1, no. 1, pp. 1–1, Jun. 2003.