

**OPC: UN ESTANDAR EN LAS REDES INDUSTRIALES
Y BUSES DE CAMPO**

Índice de Materias

Resumen:	iii
Introducción:	iv
Capítulo I: OPC (OLE Process Control)	1
1.1 Propósito.....	2
1.2 Soporte utilizado para Aplicaciones OPC.....	3
1.3 Beneficios de OPC.....	4
1.4 Visión Futura de OPC.....	5
Capítulo II: Servidores OPC	6
2.1 Servidor de Acceso a datos OPC.....	8
2.2 Servidor de Alarmas, Condiciones y Eventos OPC.....	9
2.3 Servidor de Acceso a Datos Históricos.....	10
Capítulo III: Interfaces y Arquitectura OPC	12
3.1 Interfaces OPC.....	12
3.2 Arquitectura general de OPC y sus componentes.....	12
Capítulo IV: Servidores Locales v/s Remotos	14
4.1 Uso de Servidores Locales y Remotos.....	14
4.2 Servidor OPC Browser.....	14
4.2.1 Servidores disponibles en una máquina en Particular..	14
4.2.2 Solución global.....	15
Capítulo V: Seguridad OPC	16
Capítulo VI: Conclusiones	19
6.1 Ventajas.....	19
6.2 Desventajas.....	20
6.3 Recomendaciones.....	21
Anexo A: DDE (Dinamic Data Exchange)	
Anexo B: OLE (Object Linking and Embedding)	
Glosario	
Referencias	
Bibliografía	

RESUMEN

Este trabajo está orientado a la descripción de la última tecnología y desarrollo existentes para reemplazar los sistemas propietarios de Drivers de Control y automatización, por sistemas que comprendan plataformas abiertas de sistemas operativos, mediante la utilización de OPC (Ole Process Control).

Dentro de las conclusiones orientadas a las ventajas se encuentran:

- Abaratar costos en el diseño de Software y Hardware Industrial.
- Integrar y hacer interoperable múltiples sistemas.
- Interoperabilidad en multiplataforma.

Dentro de las conclusiones orientadas a las desventajas se encuentra como la principal, que OPC presenta una débil seguridad al nivel de multiplataforma (Sistema UNIX y Windows, entre otros) por el uso de OLE/COM/DCOM.

Finalmente en este trabajo se propone un layout de la arquitectura OPC a multinivel, abarcando desde el sistema operativo hasta los sistemas de aplicación para tener un modelo de referencia en alusión al modelo ISO/OSI de TCP/IP.

Introducción

ESTADO DE ARTE

El problema histórico ha sido que los datos de diferentes sistemas tienen diferentes formatos y protocolos de comunicación. Esto ocurre, por ejemplo, cuando dispositivos son conectados a un sistema PC basado en un sistema SCADA o DCS.

Claramente el desarrollo realizado por los fabricantes de software al crear Drivers únicos para cada tipo diferente de equipo de control, no solo llevó a una duplicación de esfuerzos, sino que añade inherentemente riesgos adicionales a la consecución con éxito y a tiempo de un proyecto como así también el desarrollo de nuevo Hardware.

Es por estas razones, por lo que cinco empresas Intellution, Opto-22, Fisher-Rosemount, Rockwell Software e Intuitiv Software, junto a Microsoft, decidieron trabajar juntas para eliminar estas barreras de comunicación y definieron las interfaces basadas en **OLE/COM (Object Linking and Embedding/Common Object Model)** que sentó las bases para **OPC (Ole Process Control)**.

ALCANCE

OPC es una interfaz de programación de aplicaciones estándar para el intercambio de datos que puede simplificar el desarrollo de Drivers de I/O (Dispositivos de entrada y salida u/o Banco de Datos) y mejorar el rendimiento de los sistemas de interfaz.

Con OPC el usuario decide libremente que componente de hardware es el mejor para una determinada aplicación dado que no habrá discusiones sobre el desarrollo de Drivers. Cada fabricante de Software y Hardware solo tendrá que implementar una interfaz, no siendo necesario ya un Driver para cada sistema hardware producido por fabricantes diferentes.

PROPOSITO

Este trabajo da a conocer la Interfaz OPC a escala global, destacando sus fortalezas y debilidades.

En el **capítulo I** se estudiará el Propósito de OPC, como así también el Soporte utilizado para Aplicaciones, los beneficios de utilizarlos y la Visión Futura que se ve para él. Junto con ello se muestran las características de los Servidores OPC los que comprenden los de Acceso a Datos; Alarmas, Condiciones y Eventos; y de Acceso a Datos Históricos, todo ello en el **Capítulo II**.

En el **Capítulo III** se verá los tipos de Interface como así también la arquitectura a nivel Global. En el ámbito de arquitectura global se destacará en el **Capítulo IV** la aplicación de Servidores Locales v/s Servidores Remotos donde se discuten los inconvenientes asociados a ello, discutiendo en el **Capítulo V** los aspectos de Seguridad de Sistemas de automatización que utilizan OPC a nivel Global. Finalmente en el **Capítulo VI** se realizan las conclusiones discutiendo las ventajas y desventajas de OPC al nivel de Sistemas de control y de Red de comunicaciones.

Se destaca como información utilizada para el desarrollo de éste, las últimas especificaciones de **OPC Foundation (Fundación OPC)**, que incluye las especificaciones oficiales del año 2001 y las que se presentarán este año en Hannover, Alemania (No oficiales), como así también Boletines técnicos emanados por empresas asociadas a OPC (96 empresas) entre las que se destacan **Siemens, Rockwell Software, Keppware, Technosoftware y National Instruments**.

Capítulo I: OPC (OLE Process Control)

OPC es un mecanismo estándar de comunicación. Interconecta en forma libre numerosas fuentes de datos, donde se incluyen dispositivos de planta en la fábrica (PLC's, Variadores de Frecuencia, etc.), o un banco de datos en un cuarto de control (Dispositivos entrada/Salida (I/O)). La arquitectura de información para el Proceso Industrial se muestra en la **Figura 1.1** e involucra los siguientes niveles:

- a) Dirección Campo:** Con el advenimiento de dispositivos de campo inteligentes, del tipo SMART (Válvulas, servomecanismo, Presostatos, etc., basados en microprocesadores.), se puede integrar una gran cantidad de información, que no estaban previamente disponibles. Esta información proporciona datos del estado del dispositivo, sus parámetros de configuración, materiales de construcción, etc. Toda esta información puede ser presentada al usuario, y a cualquier aplicación que lo utilice, de una forma consistente.
- b) Dirección de Procesos:** La instalación de Sistemas de Control Distribuidos (DCS) y de Sistemas de Supervisión de Control y Adquisición de Datos (SCADA), permiten monitorear, y controlar los procesos industriales, mediante datos electrónicamente disponible, lo que antes se realizaba en forma manual.
- c) Dirección de Comercio:** Los beneficios pueden ser obtenidos instalando los sistemas de control apropiados. Esto se cumple integrando la información recolectada desde el proceso, hacia los sistemas comerciales que manejen los aspectos financieros del proceso industrial mediante Softwares especializados. De esta forma se proporciona información de una manera consistente a las aplicaciones del cliente, donde se minimizará el esfuerzo requerido para proporcionar esta integración.

Para realizar estos direccionamientos en forma eficiente, los fabricantes (Entiéndase en adelante como desarrolladores de sistemas de control y automatismos), necesitan acceder a datos "Piso-Planta" (Al interior de la

Industria) e integrarlos a sus sistemas comerciales existentes. Los fabricantes deben poder utilizar, fuera de las herramientas provistas (Paquetes SCADA, Bancos de datos, hojas de cálculo, etc.), otras para congregar "un sistema" para satisfacer sus necesidades.

La "llave" para ello, es un sistema abierto en arquitectura y de una comunicación eficaz que se concentre **en el acceso a los datos y no en el tipo de datos**.

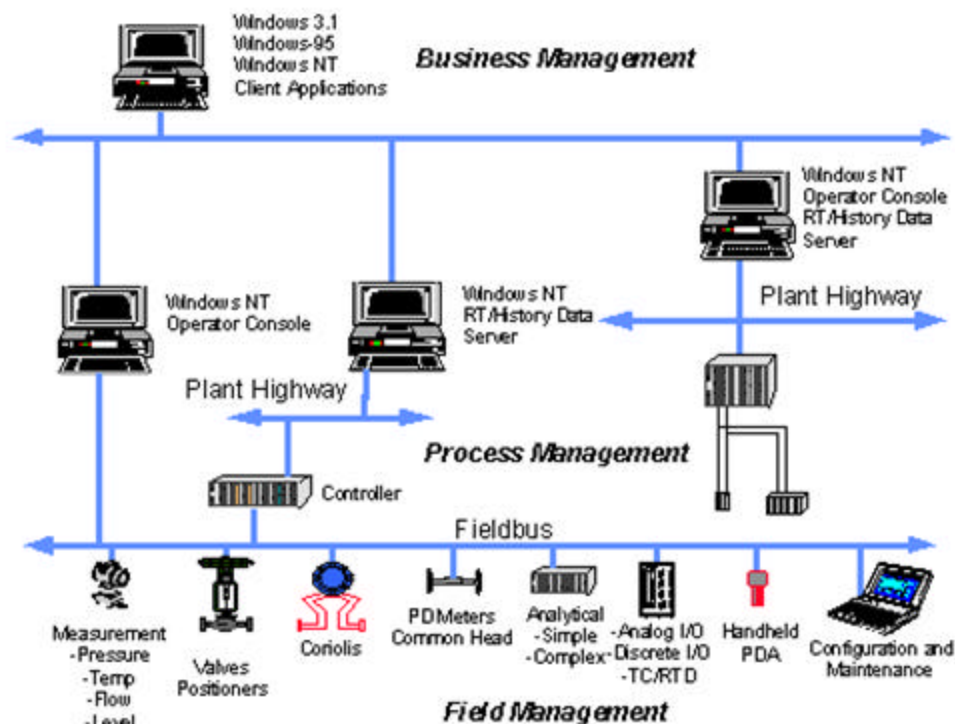


Figura 1.1: Arquitectura de información para el Proceso Industria

1.1 Propósito

Lo que se necesita es una manera de acceso común para aplicaciones específicas, con información proveniente de cualquier fuente, como lo es un dispositivo o un banco de datos.

El Servidor de OPC, de la **Figura 1.2** y en las secciones siguientes se usa como sinónimo para cualquier servidor que proporcione interfaces OPC:

- Servidor OPC de Acceso de Datos.
- Servidor OPC de Eventos y Alarmas.
- Servidor OPC de Datos Históricos.

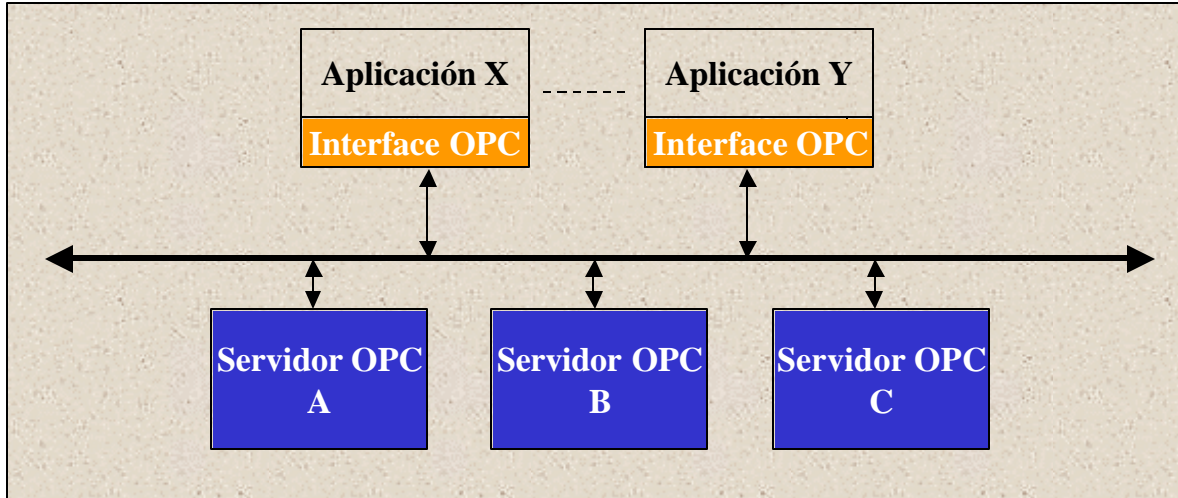


Figura 1.2: Aplicación trabajando con Varios Servidores.

OPC dibuja una línea de comunicación entre los proveedores de Hardware y los diseñadores de Software. OPC provee un mecanismo para registrar datos de una fuente de información y comunicarla a cualquier cliente, en forma transparente. Un fabricante puede desarrollar un servidor reusable (Integrando un Upgrade), favorablemente perfeccionado, que se comuniquen con la fuente de datos, y mantenga a su vez un mecanismo para acceder a la fuente de datos y dispositivos en forma eficaz. OPC proporciona una interface entre el servidor y cualquier cliente, para así poder acceder a los dispositivos.

1.2 Soporte Utilizado para Aplicaciones OPC

Casi todas las aplicaciones de Control y Supervisión están desarrolladas en ambientes de programación como lo son **Visual Basic, Delphi, Power Builder**, etc.

OPC tiene en cuenta esta tendencia y por medio de Microsoft que diseñó **OLE (Ver anexo B)** y **COM [1]** utiliza estos lenguajes como base en el diseño de aplicaciones Cliente/Servidor OPC. De esta forma se encapsulan las complejas acciones de acceso a datos, donde luego estas informaciones se pueden traspasar a planillas Excel mediante Protocolo **DDE (Dinamic Data Exchange, Ver anexo A)**, u/o representarlas mediante programas para aplicaciones específicas de control de Piso-Planta.

Basado en esta arquitectura e interfaces se planea apoyar el desarrollo de servidores OPC en otros softwares de programación basados en ambiente **LINUX o UNIX**.

1.3 Beneficios de OPC

OPC se diseñó para permitir aplicaciones donde el cliente accese a datos Piso-Planta de una manera consistente. Con aceptación por parte de los principales fabricantes, OPC proporcionará beneficios tales como:

- A) Los fabricantes de Hardware sólo tienen que desarrollar e integrar componentes al software para que los clientes (o usuarios, entiéndase por quien realiza la aplicación) los puedan utilizar en sus aplicaciones.
- B) Los diseñadores de Software no tendrán que volver a reescribir sus Drivers (Upgrade) debido a cambios de las características de su Hardware.
- C) Los Clientes tendrán más opciones de desarrollar sus sistemas de Piso-Planta, haciendo uso de la integración de una gama más amplia de sistemas de Hardware de diversos fabricantes.

Con OPC, la integración del sistema en el ambiente de la informática será más heterogénea. Con OLE/COM la distribución de sistemas como lo muestra la **Figura 1.3** se vuelve posible.

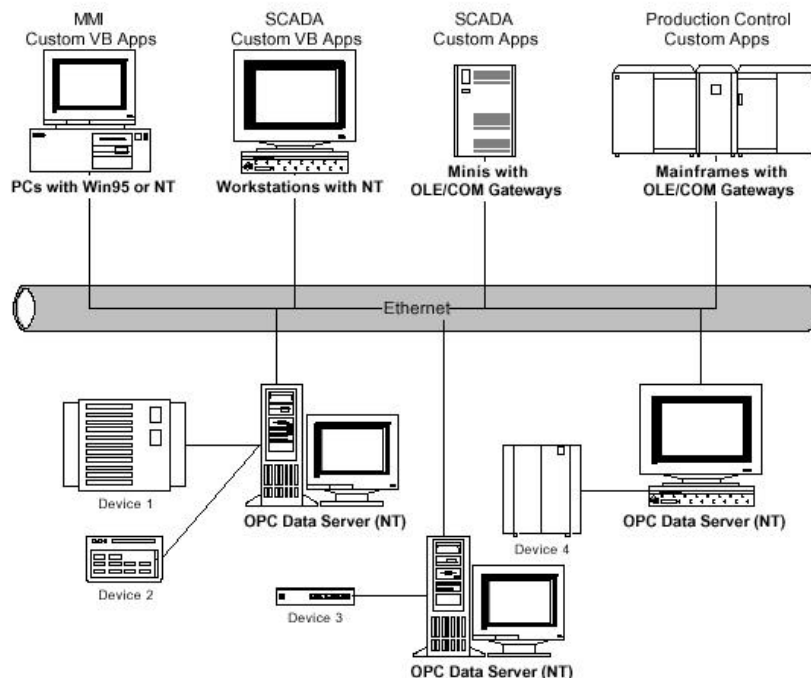


Figura 1.3: Ambiente Informático Heterogéneo.

1.4 Visión Futura de OPC

Una meta primaria para OPC es entregar especificaciones tan rápidamente como le sea posible a la industria, donde los primeros enfoques apuntan a:

- a)** Acceso a datos "On Line", es decir, la lectura y escritura eficaz de datos entre una aplicación y un dispositivo de control de proceso, en forma flexible y eficaz.
- b)** Manejo de "Alarmas y Eventos", es decir, la existencia de mecanismos de notificación de ocurrencias de eventos específicos y condiciones de alarma.
- c)** Acceso a Datos Históricos, es decir, procesamiento y revisión de datos, lo que implica análisis de tendencias.
- d)** Funcionalidad como a la seguridad, acceso a datos "On Line", Alarmas Histórica y a Datos Históricos en forma eficaz y con un nivel de seguridad óptimo.

La arquitectura OPC se ve influenciada por las ventajas de COM, el cual proporciona un mecanismo conveniente para extender la funcionalidad de OPC. Otras metas a alcanzar mediante el diseño de OPC son:

- a)** Ser un instrumento simple de Control.
- b)** Ser flexible a las necesidades de múltiples fabricantes.
- c)** Proporcionar un alto nivel de funcionalidad.
- d)** Permitir un funcionamiento eficaz.

Las características técnicas que ofrece la OPC Foundation (Fundación OPC) a los fabricantes de dispositivos de control y sistemas DCS y SCADA incluyen:

- a)** Un set de características que la interface COM ofrece para el uso de clientes y servidores.
- b)** Un set de referencias de la interface de automatización OLE, para apoyar desarrollos, al nivel de clientes con aplicaciones comerciales de alto nivel como lo son Excel, Visual Basic, Power Builder, etc.

Capítulo II:

Servidores OPC

Un Cliente OPC puede conectarse, por medio de una red a Servidores OPC proporcionados por uno o más fabricantes [2]. De esta forma no existe restricción por cuanto a tener un Software Cliente para un Software Servidor, lo que es un problema de interoperabilidad que hoy en día se aprecia con sistemas del tipo propietario. Lo anterior se puede apreciar en la **Figura 2.1**.

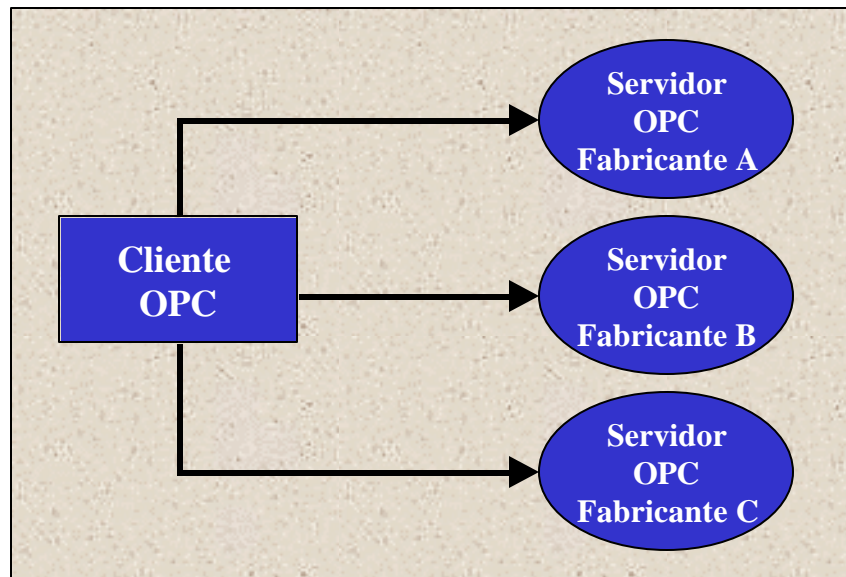


Figura 2.1: OPC Cliente.

Los fabricantes, a su vez, proporcionan el código que identifica: Dispositivos, Tipos de Datos a los que cada servidor tiene acceso, Valor de los Datos, y detalles sobre cómo el Servidor físicamente accesa a los datos. Sin estos códigos Servidores y Clientes no podrían comunicarse y reconocerse como sistemas compatibles.

La **Figura 2.2**. muestra la interoperabilidad de diversos Sistemas interconectados dentro un una misma Red si solo si estos trabajan bajo un estándar OPC.

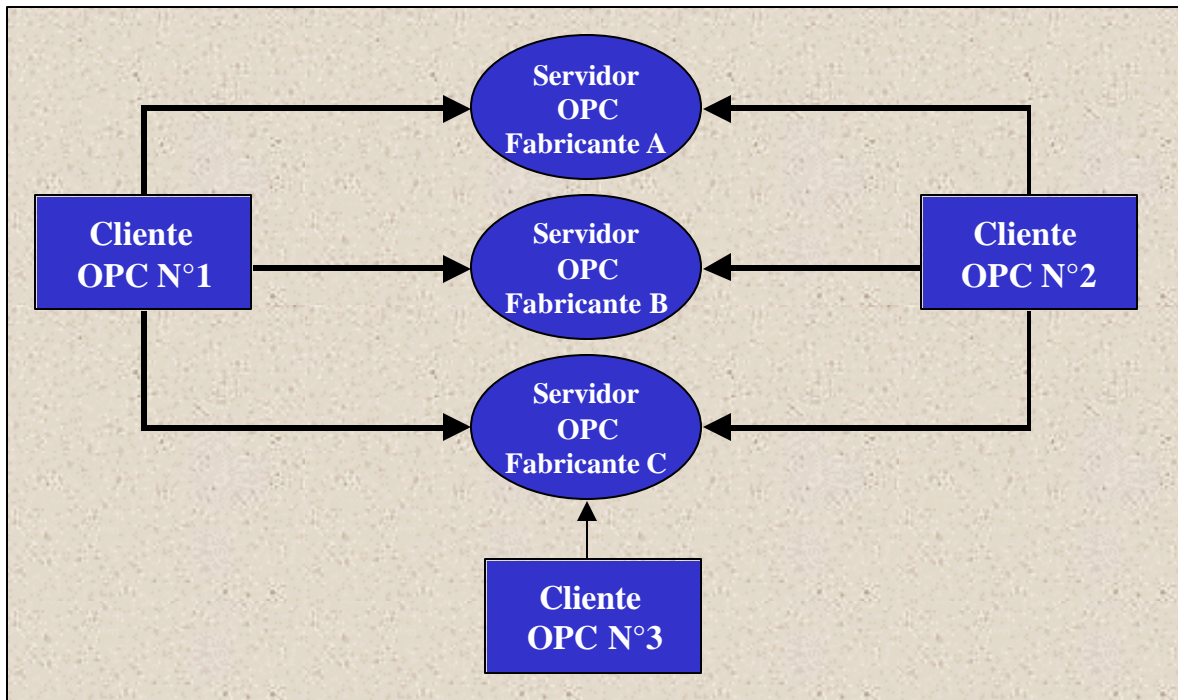


Figura 2.2: Relación de trabajo Cliente/Servidor.

También es posible que otros sistemas como lo son SCADA o DCS puedan comunicarse con un Servidor OPC [2] y llevar su información recopilada desde un banco de datos o dispositivos físicos como lo son del tipo SMART y PLC's. Así de esta forma aplicaciones cliente OPC de otros fabricantes tendrán acceso a estos datos por medio del Servidor. Lo anterior se observa en la **Figura 2.3**.

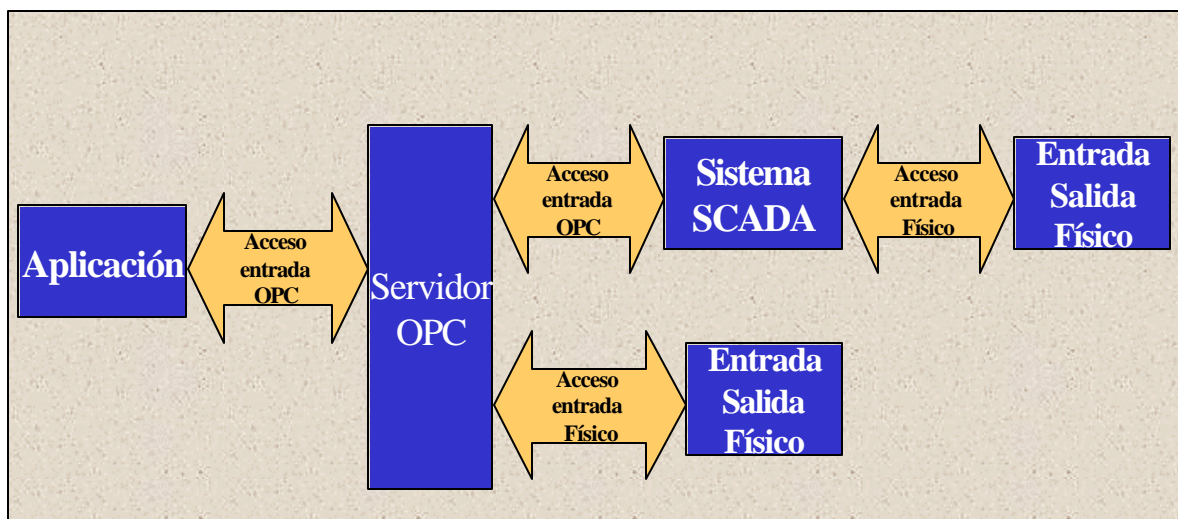


Figura 2.3: Relación Cliente Servidor y Sistemas SCADA.

A continuación, se presenta una visión global del funcionamiento de los Servidores OPC de Acceso a Datos; de Alarmas, Condiciones y Eventos como así también de Acceso a Datos Históricos.

2.1 Servidor de Acceso a datos OPC

A un alto nivel, un Servidor de Acceso a Datos OPC, se compone de varios objetos: **Servidor, Grupo, e Item**. La función del servidor OPC, es mantener la información sobre sí mismo y hacer las veces de un "Recipiente" unificando los datos en un Grupo. La función del Grupo OPC es mantener la información y proporcionar un mecanismo por contener y organizar lógicamente los Items.

Los Grupos OPC proveen a los clientes OPC, quienes ejecutan aplicaciones, una forma de organizar sus datos. Por ejemplo, el grupo podría representar los Items de un dispositivo en particular para que despliegue o informe sobre sus datos. Pueden leerse datos y pueden escribirse.

Basado en conexiones excepcionales también pueden crearse conexiones entre el cliente y los ítems en el grupo y pueden habilitarse y desactivarse como se necesite. Un cliente OPC puede configurar la tasa de trasferencias de servicio de su servidor OPC, en cuanto a proporcionar los cambio de datos que se presenten.

Hay dos tipos de grupos, Público y Local [2], que se describen tal como sigue:

- a) **Público:** Es compartido por múltiples clientes. Hay también interfaces optativas específicas para grupos públicos en plataforma Linux o Unix [3]
- b) **Local:** Trabaja en torno a un cliente o grupo con prioridad.

Dentro de cada Grupo el cliente puede definir uno o más Artículos de OPC. Lo anterior se observa en la **Figura 2.4**.

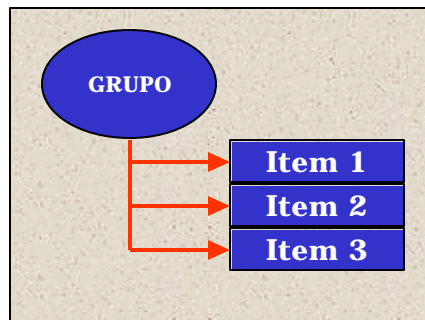


Figura 2.4: Relación de trabajo Grupo/Item.

Los Items OPC representan conexiones a las fuentes de datos dentro del servidor. Un Item OPC, bajo la perspectiva de interface, no es accesible como un objeto por un Cliente OPC. Por consiguiente, ninguna interface externa se encuentra definida para un Item OPC. Todos acceden a los Items OPC vía Grupo OPC, objeto o ícono que “contiene” el (los) Item(es) OPC, o simplemente donde el ítem OPC se define.

Asociando, un Item es un valor, una condición y permanece o varía en el tiempo. El valor está en la forma de una variable, y la condición es similar a lo especificado por Fieldbus (Estándar de Buses de Campo).

Note que los Items no son las fuentes de los datos; ellos son sólo conexiones a ellos. Por ejemplo, los Tags (etiquetas) en un sistema DCS existen sin tener en cuenta si un cliente OPC está accediéndolos o no durante su funcionamiento. Los Items OPC deben pensarse simplemente como la dirección específica de los datos, no como la fuente física real de los datos que referencia la dirección.

2.2 Servidor de Alarmas, Condiciones y Eventos OPC

Estas interfaces proveen de mecanismos a los Clientes OPC, con los cuales pueden ser notificados de la ocurrencia de eventos y condiciones de alarmas específicas [4]. Estas también proporcionan servicios que les permiten a los Clientes OPC determinar eventos y condiciones necesarias para alarmas, eventos, y para obtener su estado actual, todo ello apoyado por un Servidor OPC. Dentro de OPC se puede definir que:

- a) Alarma:** Es una condición anormal del sistema y por lo que es un caso especial de esta.
- b) Condición:** Es un estado nombrado Evento en el Servidor OPC. Por ejemplo, la etiqueta FC101 puede tener las condiciones siguientes asociadas con ella:
 - HighAlarm, HighHighAlarm,
 - Normal, LowAlarm,
 - LowLowAlarm.
- c) Evento:** Es una ocurrencia perceptible que es de importancia al Servidor OPC, de los dispositivos que representa o sus Clientes OPC.

Un evento puede o no ser asociado con una condición. Por ejemplo, la transición de HighAlarm a condiciones normales es un evento. Sin embargo, una acción del operador permite cambiar la configuración del

sistema, y los errores son ejemplos de eventos que no se relacionan a las condiciones específicas del sistema. Los Clientes de OPC pueden suscribirse al sistema para ser notificados de las ocurrencias de eventos específicos.

La interface de **IOPCEventServer** [4], proporciona los métodos que habilitan al Cliente OPC a:

- a) Determinar los tipos de eventos que los Servidores OPC soportan.
- b) Suscribirse a eventos específicos, para que puedan recibir notificaciones de sus intervalos de ocurrencias. Pueden usarse filtros para definir un subconjunto de eventos deseados.
- c) Accesar y manipular las condiciones asignadas al Servidor OPC.

Además de la interface de IOPCEventServer, un Servidor OPC puede apoyar interfaces optativas para revisar condiciones llevadas a cabo por el servidor y para los grupos con propiedad de condición Pública.

2.3 Servidor de Acceso a Datos Históricos

Aunque OPC se diseñó en un principio para acceder datos a un servidor de Red, se vio que las interfaces OPC pueden ser usadas en muchas formas y lugares dentro de una aplicación. En el nivel más bajo, este puede acceder a datos proporcionados por dispositivos físicos en forma directa y/o en aplicaciones dentro de un sistema SCADA o DCS tal como se vio en la **Figura 2.3**.

La arquitectura y el diseño de la especificación **Historical Data Access Automation Interface Standard**, de la Fundación OPC, hace posible que un Servidor OPC, permita activar una aplicación cliente y acceder a datos de muchos Servidores OPC u otros sistemas OPC, proporcionados por diferentes fabricantes.

Lo anterior es posible gracias a la interface **OPC Historical Data Access (OPCHDA)** [5] la cual se apoya en los lenguajes Visual C++ y Visual Basic. Ellos permiten que cualquier aplicación en control y automatización con OLE puedan acceder a Interfaces OPC, según las limitaciones de la aplicación respectiva.

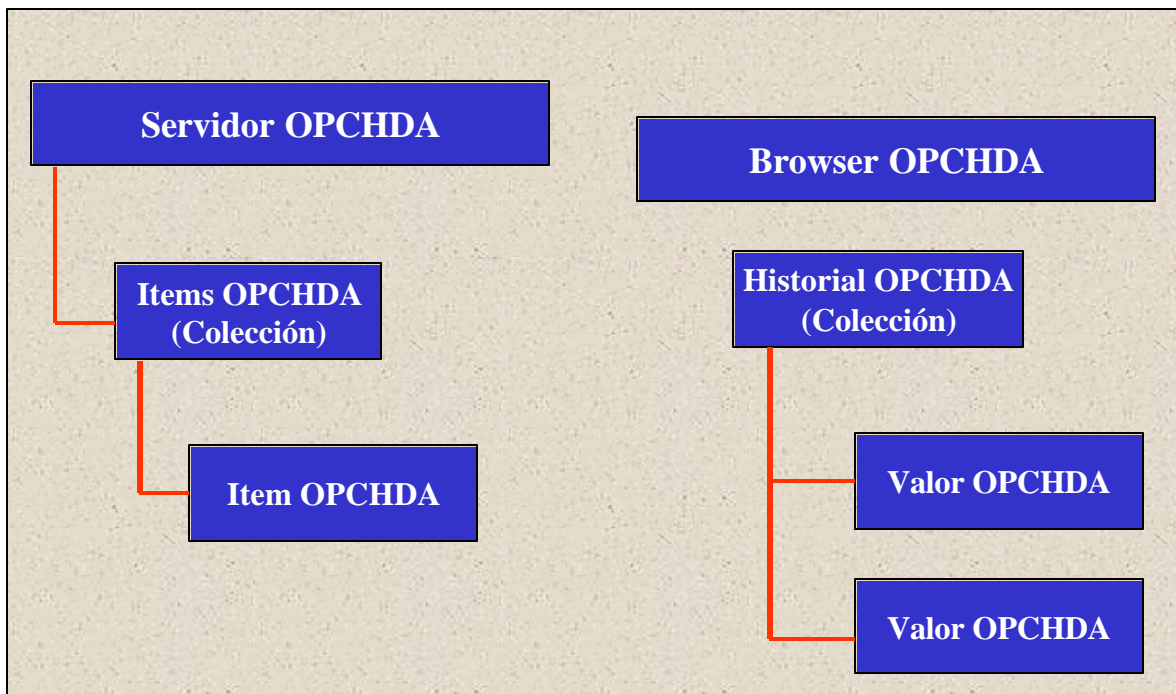


Figura 2.5: Acceso a valores y Datos Históricos mediante OPCHDA.

Un cliente OPC conecta a un Servidor de Automatización OPC que se comunica a la fuente de datos subyacente (Por ejemplo un servidor OPC HDA) mediante interfaces funcionales OPC (OLE/COM).

El servidor OPCHDA, proporciona una manera de acceder o de comunicarse a una fuente de datos históricos (Mediante lectura u/o escritura). Los tipos de fuentes disponible dependen de la función de trabajo del servidor. Junto con ello se permite que otros servidores mediante aplicaciones Browser puedan acceder a los datos.

El Servidor OPCHDA proporciona una colección de ítems compuesta de objetos de automatización OPCHDA El Item OPCHDA proporciona una conexión a un solo artículo de datos en la fuente de los datos subyacente.

Paralelamente al acceder a los Items OPCHDA, se obtiene acceso a los datos Históricos y sus valores en el tiempo, siendo esto ultimo programable mediante una aplicación en el servidor. Todo lo anteriormente descrito se puede apreciar en la **Figura 2.5.**

3.1 Interfaces OPC

Las características técnicas de OPC contienen siempre dos juegos de interfaces; Interface diseñada para un propósito (Aplicación) y una Interface de Automatización[2; 6]. Esto se muestra en **Figura 3.1**.

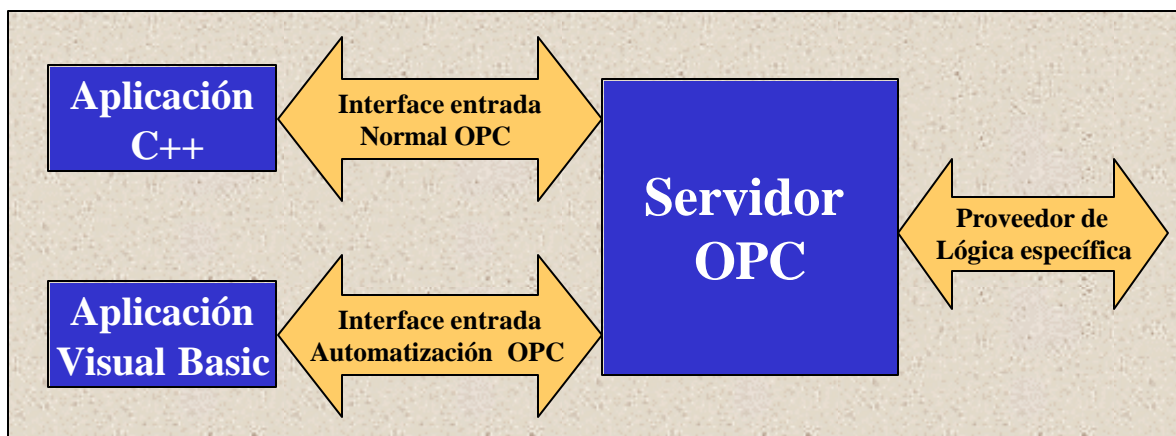


Figura 3.1: Interfaces OPC.

OPC especifica la interface COM [1] (Component Object Model), **lo que la interface es y su aplicación y no su implementación**. Especifica el comportamiento esperado que proporciona la interface ante el uso y/o aplicaciones del cliente. Comprende tener las descripciones de arquitecturas e interfaces. Como todas las aplicaciones COM, la arquitectura OPC es un modelo Cliente-Servidor donde el componente Servidor OPC, proporciona una interface con el objeto y lo controla.

3.2 Arquitectura General de OPC y sus Componentes

Hay varias consideraciones, que son únicas, para llevar a cabo la implementación de un servidor OPC [2]. Una de ellas, la principal, es la **frecuencia de traslado e intercambio de datos** a través de redes comunicacionales, hacia dispositivos físicos u otras bases de datos las cuales son incompatibles entre sí. De esta manera, se espera que

Servidores OPC sean ejecutables ya sean en forma Local o Remota, los cuales incluyan un código que los identifique y los respalde en la recolección de datos en forma eficaz entre un dispositivo físico o una base de datos.

Una aplicación Cliente OPC se comunica con un Servidor OPC a través de un cliente específico e interfaces de automatización. Los servidores OPC deben llevar a cabo la interface del cliente, y opcionalmente puede llevar a cabo la interface de automatización, tal como lo describe la arquitectura típica OPC, de la **Figura 3.2**.

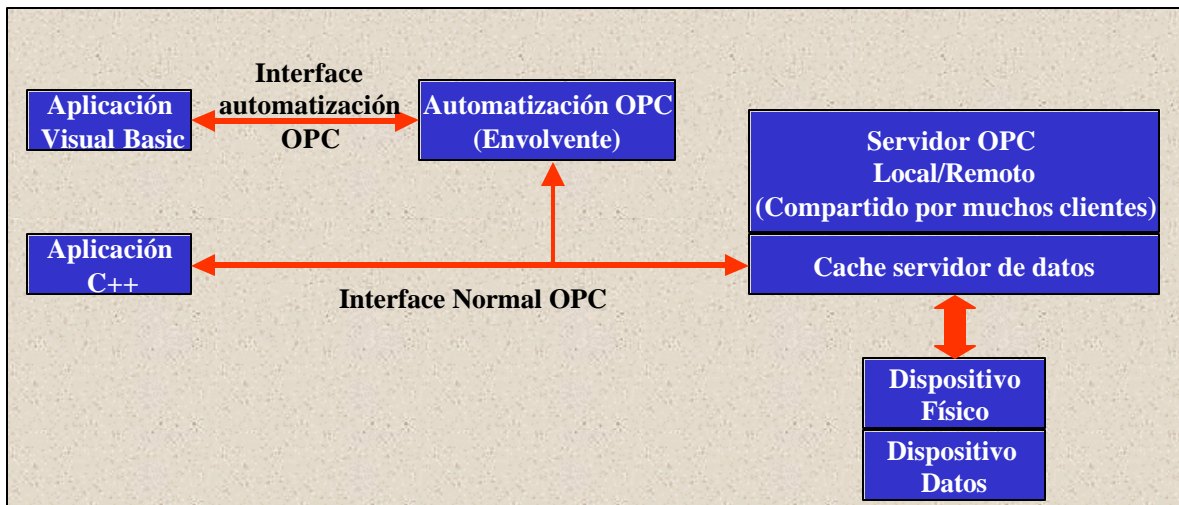


Figura 3.2: Arquitectura Típica OPC.

En algunos casos la Fundación OPC proporciona una interface estándar de automatización. Esta interface, denominada **wrapperDLL [6]**, puede usarse para cualquier Cliente-Servidor o fabricante específico.

Capítulo IV

Servidores Locales v/s Remotos

4.1 Uso de Servidores Locales y Remotos

Se espera que los fabricantes de servidores OPC, tomen en cuenta uno de los dos puntos que se presentan a continuación, al conectar una red de computadoras:

- a) Pueden indicar que el cliente siempre debe conectarse a un servidor local, al que hará uso de un esquema de red del tipo propietario. Este acercamiento normalmente será usado por fabricantes que agreguen OPC a un producto DCS o SCADA existente.
- b) Pueden indicar que el cliente debe conectarse al servidor deseado, en el nodo designado, y hacer uso de **DCOM (Distributed Component Object Model)** [7] para proveer conexión a una Red mediante la interface **IOPCServerList**.

4.2 Servidor OPC Browser

La Interface del Servidor OPC Browser (Buscador de Servidores OPC) **IOPCServerList** se especifica como parte del documento **OPC Common Definitions and Interfaces** [6] de la Fundación OPC.

4.2.1 Servidores disponibles en una máquina en Particular

Los servidores OPC se registran en el sistema mediante una categorización de los componentes vía **ICatInformation (IID_ICatInformation)** [4] de Microsoft. Esto determina que interfaces en el **StdComponentCategoriesMgr (CLSID_StdComponentCategoriesMgr)** [4] deben ser usadas para determinar qué servidores OPC se encuentran instalados en la máquina local.

El problema es que esto no es aplicable para las máquinas remotas porque la administración de la categorización de los componentes son **DLL**

(Dynamic Link Library) y la interface ICatInformation sólo trabaja **In-Proceso**. Como resultado de esto, no es fácil para el cliente obtener una lista de Servidores OPC instalados en una máquina remota.

4.2.2 Solución global

La Fundación OPC provee el Servidor Browser **OPCENUM.EXE [6]** el cual puede residir en una máquina cualquiera, el cual accede a la raíz de las Categorías de Componentes locales y proporciona una nueva interface **IOPCServerList** que puede ordenarse y ser usado por clientes remotos.

Este servidor tiene un **CLSID** (Clase de Direccionamiento) publicado y puede instalarse una vez dentro de cualquier máquina que contenga un Servidor OPC. El cliente todavía necesita saber el nombre del nodo de la máquina designada, sin embargo él puede crear ahora remotamente este objeto y puede usar el **IOPCServerList**, que determina que tipos servidores y marcas están disponibles en esa máquina.

Capítulo V

Seguridad

La seguridad en servidores OPC, es consistente con el modelo de seguridad Windows NT, dando a los usuarios (Clientes) igual tipo de restricciones otorgadas a clientes NT. La **Figura 5.1** muestra un modelo que sirve de estructura conceptual para la especificación **OPC Security Custom Interface [8]**.

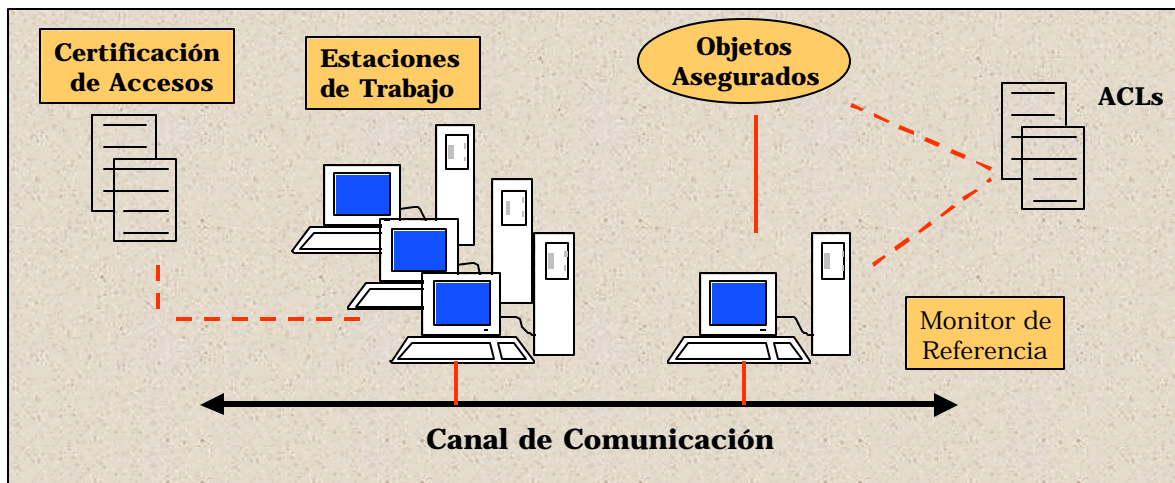


Figura 5.1: **Modelo de seguridad Cliente/Servidor OPC**

Cada estación de trabajo posee un certificado de acceso (Access Token) para trabajar, apoyado por DCOM, dentro del canal de comunicación. Al solicitar al Monitor de referencia acceso a los Objetos Asegurados (con restricción), este verificará su nivel de acceso con el Control de Lista de Acceso (ACL). El monitor de referencia no tiene acceso a Leer las comunicaciones en el Canal de Comunicación.

Un servidor OPC puede llevar a cabo tres niveles de seguridad que son:

- a) **Seguridad Inválida:** No existe patrones de seguridad entre los OPC clientes y el OPC Servidor. Todos tienen por igual acceso al servidor.

- b) **Seguridad DCOM:** Limita el acceso y atributos sobre el servidor OPC por parte de clientes seleccionados. Estos pueden ser, por ejemplo, permisos de acceso para aplicaciones clientes. Sin embargo, el Servidor OPC no controla totalmente el acceso a su sistema por parte de cualquier otro Sistema operativo (Linux y Unix) a los objetos con restricciones específicos.
- c) **Seguridad OPC:** El Servidor OPC sirve como un regulador de control de acceso a fabricantes de sistemas operativos como Linux y Unix sobre objetos específicos de acceso restringido que son expuesto por el Servidor OPC. Un Servidor OPC puede llevar a cabo Seguridad OPC además de Seguridad de DCOM, o Seguridad OPC en forma exclusiva.

Siempre que sea posible un Servidor OPC llevará a cabo Seguridad OPC basado en decisiones de acceso mediante la utilización y autorización de un **NT Access Token** (accesar al servidor, en red, mediante una "ficha") asociada con la aplicación cliente. Este acercamiento permite seguridad, dado que es transparente a las aplicaciones cliente, porque lo que no hay ninguna necesidad de tomar acciones explícitas para establecer un certificado de acceso adicional. También, porque NT Access Token es independiente del Servidor OPC, siendo esto más fácil para escribir aplicaciones cliente portátil.

Sin embargo hay circunstancias que evitan el uso de **NT Access Token**. Algunos ejemplos de tales situaciones son:

- El Servidor OPC está aplicándose delante de un sistema operativo distinto de Windows NT que apoya DCOM como Linux y UNIX.
- Las aplicaciones del cliente pueden estar corriendo en un sistema operativo que no apoya la creación de **NT Access Token**
- El Servidor OPC y las aplicaciones cliente son distribuidas en dispositivos múltiples fuera del contexto del dominio de la red.

Para tales situaciones, la especificación **OPC Security Custom Interface** de Fundación OPC define la condición para el Servidor OPC de poder llevar a cabo una credencial privada (certificado de acceso) que realiza un proceso de verificación.

Un ejemplo podría ser el uso, por parte de un servidor de un **UserID** (Identificación de Usuario) específico, en combinación con una **Pass (Contraseña)**. Depende del Servidor de OPC usar mecanismos que

prevengan el compromiso de estas credenciales. Por ejemplo, no deben transmitirse contraseñas en texto claro, ni ellos deben guardarse en disco en texto claro.

Capítulo VI

Conclusiones

Sin duda alguna, OPC presenta una gran solución a los problemas de Interoperabilidad e Integración en torno a sistemas de control y de supervisión industrial.

El hecho que OPC se sustente en la plataforma OLE/COM de Microsoft y considerando que la mayoría de las aplicaciones industriales se encuentran realizadas en Power Builder, Visual C++ y Visual Basic, no es raro notar que el estándar OPC lo hayan adoptado más de Noventa y Seis fabricantes de los Cinco que fueron originalmente.

También se destaca la posibilidad de integrar OPC con otros sistemas operativos como lo son Linux y Unix, dejando la posibilidad a los integradores de trabajar en otros sistemas que consideren más eficientes o a los cuales hayan desarrollado nuevas aplicaciones.

A continuación se desprenden las ventajas y desventajas competitivas de utilizar OPC en los Sistema de Control y de Supervisión.

6.1 Algunas ventajas de implementar OPC a nivel Industrial son:

- a)** Integración de distintas tecnologías de diferentes fabricantes dentro de un mismo sistema. *La industria no tendrá que trabajar con un solo Sistema Propietario o Sistemas SCADA o DCS específicos.*
- b)** Costos de desarrollo de sistemas de aplicación menores, dado que se está trabajando en una plataforma universal (OLE/COM), por lo que se evita duplicidad de esfuerzos. *La adquisición de sistemas de aplicación y de Upgrade será más económica y además se podrá desarrollar sistemas específicos, según nuestras necesidades y no las que nos ofrezcan los fabricantes.*
- c)** Permite integración de múltiples plataformas (Windows, Linux, Unix, suSe) mediante la utilización de COM, DCOM, Active X y Entire X. *Permite un desarrollo comunicacional dentro de redes LAN (Local Access Network) y WAN (World Access Network),, como así también exportar datos a Internet.*

- d) Comunicación On line expedita, eficaz y flexible desde el nivel de procesos hasta el nivel de Gestión. *Por medio de Softwares industriales específicos se puede lograr un mayor control dentro del proceso productivo, y optimizar Materias Primas, Recursos, Costos, etc.*
- e) Del presente trabajo, basado en las especificaciones dadas por la Fundación OPC y otros recursos bibliográficos, se propone el modelo de la **Figura 6.1** para representar OPC dentro de un sistema que va desde el nivel de aplicaciones, hasta el nivel del Sistema operativos Múltiples.

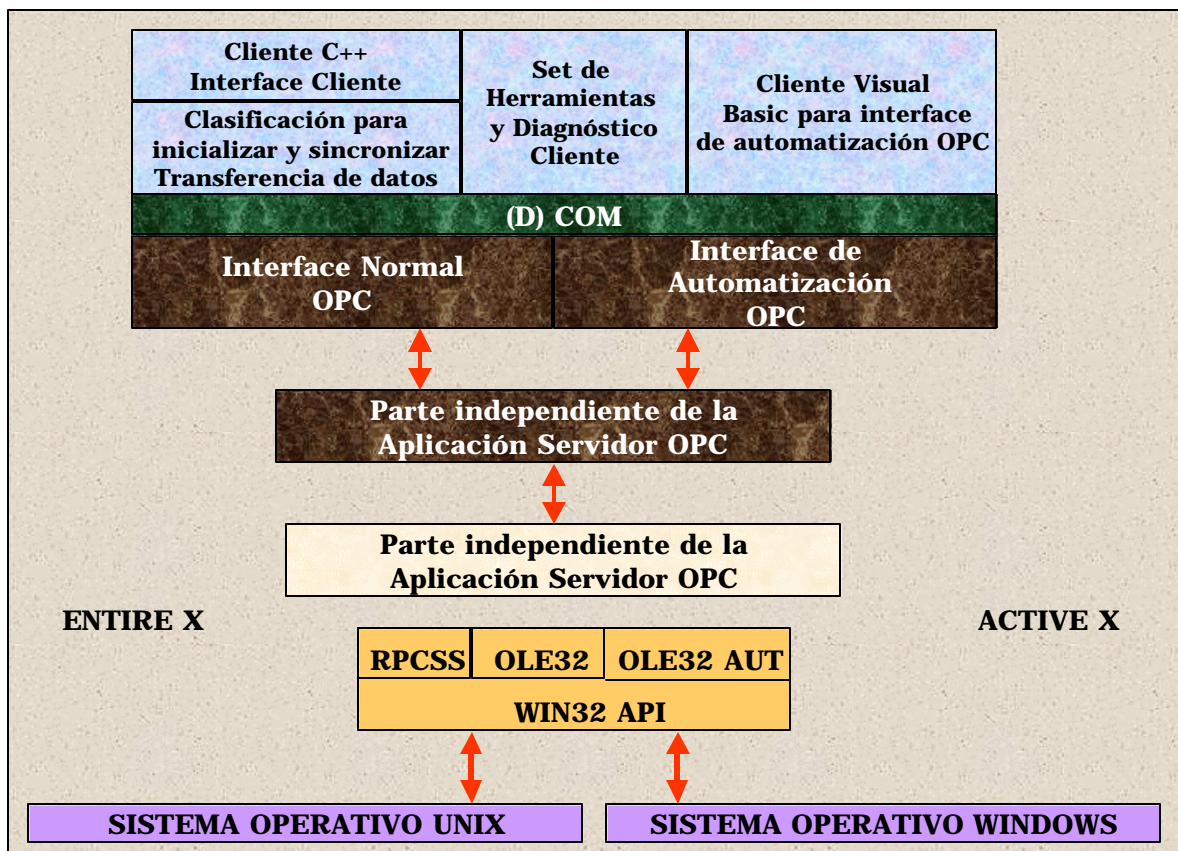


Figura 6.1: Modelo OPC

6.2 Desventajas de utilizar OPC en Sistemas Industriales

OPC parece ser un sistema industrial ideal, pero al ser tan transparente en el ámbito de las aplicaciones como así también interoperable en distintas plataformas presenta inherentemente problemas de Seguridad en estos aspectos. Un ejemplo de ello son:

- a) Los fabricantes de Buses de Campo (Fieldbus), han desarrollado Gateways para interconectar sus protocolos propietarios a redes Ethernet. *Personas inescrupulosas podrían acceder desde el nivel de gestión hasta el nivel de proceso, dañando potencialmente todo el sistema comunicacional.*

- b) Hoy en día existe la capacidad de desarrollar programas ejecutables en base a OLE/COM/DCOM, e ingresar a sistemas de red de diferentes sistemas operativos a nivel WAN, mediante Entire X y Active X. *La red de comunicaciones OPC puede estar expuesta a escala mundial y accesible desde cualquier plataforma.operativa.*

6.3 Recomendaciones

Se recomienda acrecentar los niveles de seguridad, usando Softwares de Encriptación de datos (Para él tráfico de Datagramas) y FireGuards (Cortafuegos) para proteger la Red de datos de otras externas.

GLOSARIO

ACTIVE X: Plataforma integral y abierta, utilizada para aplicaciones portátiles y de contenidos interactivos para la World Wide Web o la llamada también Tela de araña Mundial.

API: Interface de aplicación de programas, Conjunto de rutinas del sistema que se pueden usar en un programa para la gestión entrada/salida, gestión de ficheros etc.

BROWSER: Termino aplicado normalmente a los programas que permiten acceder al servicio WWW.

COM: Modelo de Componentes Objetos, proporciona interfaces y comunicación entre los componentes. A través de COM, una aplicación puede usar rasgos de cualquier otro objeto de la aplicación. COM es el centro de DCOM, ActiveX y OLE.

DCOM: La Distribución de Modelos de Componentes Objetos extiende a COM para trabajar encima de una red. Es un protocolo donde los componentes remotos parecen ser locales. El puerto DCOM hace posible integrar otros sistemas operativos como lo son Linux y Unix.

DLL: Son librerías de procedimientos residentes en ficheros. Contienen códigos, datos y recursos de Windows, que se enlazan a las aplicaciones en tiempo de ejecución. Son accesibles desde cualquier punto de un programa.

ENTIRE X: Plataforma de integración entre cualquier tipo de aplicación, facilitando la explotación en entornos distribuidos o en la Web. Con Entire X es posible resolver cualquier problema de comunicaciones, acceder a múltiples fuentes de información y administrar, de forma flexible y centralizada, entornos heterogéneos multiplataforma, manteniendo la integridad y la seguridad del sistema de información corporativo.

LINUX: Versión Shareware del conocido sistema operativo UNIX. Es un sistema multitarea multiusuario de 32 bits para PC.

OLE: Este protocolo tiene la capacidad de insertar objetos de distintos tipos en documentos de otro tipo.

OLE32: Librería OLE 32 bits.

OLE32AUT: Librería de automatización OLE 32bits.

OPC: Define un set estandarizado de interfaces, propiedades y métodos para el uso en procesos de Control y aplicaciones de software en el área de la automatización.

NT: De NET (Red), alusivo a sistemas Windows NT, UNIX, otros.

RPCSS: Servicio de Control Manager.

UNIX: Sistema operativo multitarea, multiusuario. Gran parte de las características de otros sistemas más conocidos como MS-DOS están basadas en este sistema, muy extendido para grandes servidores.

WIN 32 API: Contiene un sistema de librería de llamado WIN32, el Kernel del proceso de emulación del Windows NT, y el subsistema de autenticación.

REFERENCIAS

- [1] Microsoft Corporation, **The COM Specification**, Copyright 1995.
- [2] OPC Foundation, **OPC Data Access Custom Interface Specification V2.02**, 27 de Julio de 1999.
- [3] Technosoftware, **Software Solutions for OPC Server, Client and Diagnostic Tools**, Mayo del 2001.
- [4] OPC Foundation, **OPC Alarms and Events V1.02**, 02 de Noviembre de 1999.
- [5] OPC Foundation, **OPC Historical Data Access Automation Specification**, V1.0, 26 de Enero del 2001.
- [6] OPC Task Force, **OPC Common Definitions**, V1.0, 27 de Octubre de 1998.
- [7] Microsoft Corporation, **DCOM Technical Overview**, Redmond WA 98052-6399, 1996
- [8] OPC Foundation, **OPC Security Custom Interface**, V1.0, 17 de Octubre del 2000.
- [9] Dorin Carstoiu, Ana Brodschi, Codrut Eftimiu, **Netware Dinamic Data Exchange**, 0-7803-1772-6/94, IEEE1994.
- [10] Kraig Brockshmidt, **Inside OLE**, Microsoft Press, redmond, WA, 1995.

BIBLIOGRAFIA

- 1.- Technosoftware, **Software Solutions for OPC Server, Client and Diagnostic Tools**, Mayo del 2001.
- 2.- OPC Foundation, **OPC Batch Automation OPC Interface Specification**, V1.0, 19 de Julio del 2001.
- 3.- OPC Foundation **OPC Batch Specification**, V2.0, 19 de Julio del 2001.
- 4.- Technosoftware, "**Software Solutions for OPC Server, Client and Diagnostic Tools**", Mayo del 2001.
- 5.- OPC Foundation, **OPC Historical Data Access Automation Specification**, V1.0, 26 de Enero del 2001.
- 6.- Don Holley, **The Industrial Book**, 2001, "OPC DX Glues Fieldbuses Together", N° 8, Editor Bill George, Reino Unido,.
- 7.- Jonas Berge, **The Industrial Book**, 2001, "An issue of system architecture", N°8, Editor Bill George, Reino Unido,.
- 8.- Paul Wacker & Mark Hensley, **The Industrial Book**, 2001, "Extending Connections", N°8, Editor Bill George, Reino Unido, 2001.
- 9.- OPC Foundation, **OPC Batch Custom Interface Specification**, V 1.0, 24 de Abril del 2000.
- 10.- OPC Foundation, **OPC Security Custom Interface**, V1.0, 17 de Octubre del 2000.
- 11.- OPC Foundation, **OPC Data Access Custom Interface Specification**, V2.04, 05 de Septiembre del 2000.
- 12.- OPC Foundation, **OPC Alarms and Events**, V1.02, 02 de Noviembre de 1999.
- 13.- OPC Foundation, **OPC Data Access Custom Interface Specification**, V2.02, 27 de Julio de 1999.
- 14.- Jon Bates & Tim Tomkins, **Descubre Visual Basic C++ 6**, Prentice Hall, Madrid, 1999.

- 15.- OPC Task Force, **OPC Common Definitions**, V1.0, 27 de Octubre de 1998.
- 16.- OPC Foundation, **OPC Specification**, V1.0A, 30 de Julio de 1997.
- 17.- Microsoft Corporation, **DCOM Technical Overview**, Redmond WA 98052-6399, 1996
- 18.- Kraig Brockshmidt, **Inside OLE**, Microsoft Press, redmond, WA, 1995.
- 19.- Microsoft Corporation, **The COM Specification**, Copyright 1995.
- 20.- Dorin Carstoiu, Ana Brodschi, Codrut Eftimiu, **Netware Dinamic Data Exchange**, 0-7803-1772-6/94, IEEE1994.