CIMug Meeting – 10 June 2008
Västerås, Sweden
GID Tutorial

Generic Interface Definition (GID) Tutorial

Ralph Mackiewicz, Vice President
SISCO, Inc.
6605 19½ Mile Road
Sterling Heights, MI 48314 USA
Tel: +1-586-254-0020 x103
Fax: +1-586-254-0053
E-Mail: ralph@sisconet.com

Acronyms

- Acronyms are unavoidable when discussing integration technology.
- It was my objective to define all acronyms before using them. Please excuse any oversights.
- If you are not certain, please ask a question immediately.

Definitions:

Interoperability

The ability of computer systems to exchange information with other systems.

Integration

The ability of computer based applications to interact with other systems in order to perform a useful function for the user.

Interoperability and Integration

- Easy to Achieve:

A Better Way

- Interoperability and Integration without having to program it all yourself:
  - Where applications and devices are **inherently** capable of interoperating with other systems and performing integrated application functions in a cooperative and distributed manner.
- This is only possible if there are standards to enable it.
  - This work is progressing.
  - This is what IEC TC57 working groups are all about

CIMug Meeting – 10 June 2008
Västerås, Sweden
GID Tutorial

Web Services and Interface Standards:

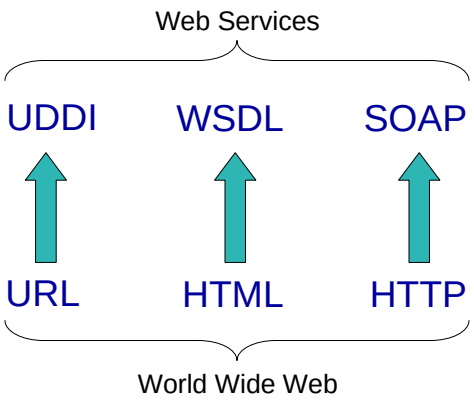
What are they and why we need them

What are Web Services?

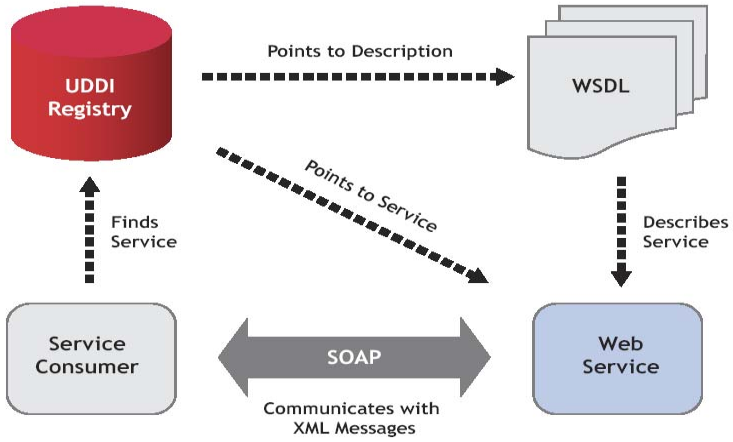
- A structured integration architecture using XML messaging over a similar network architecture as the world-wide web (WWW)
- Web Services:
  - Can reuse existing network infrastructure used by the WWW (Internet, Intranet)
  - Allows a client to discover the services and data supported by a server.
  - Defined by the World Wide Web Consortium: http://www.w3c.org
  - Uses a standard framework for integration development using widely deployed tools (Java (etc.), .Net, etc.)

Web Services is Based on the WWW

- World Wide Web (WWW)
  - Uniform Resource Identifiers (URI) or locator (URL)
  - Hyper-Text Markup Language (HTML) and the eXtensible Markup Language (XML)
  - Hyper-Text Transport Protocol (HTTP)
- Web Services
  - Universal Description, Discovery, and Integration (UDDI)
  - Web Services Description Language (WSDL) using XML
  - Simple Object Access Protocol (SOAP)



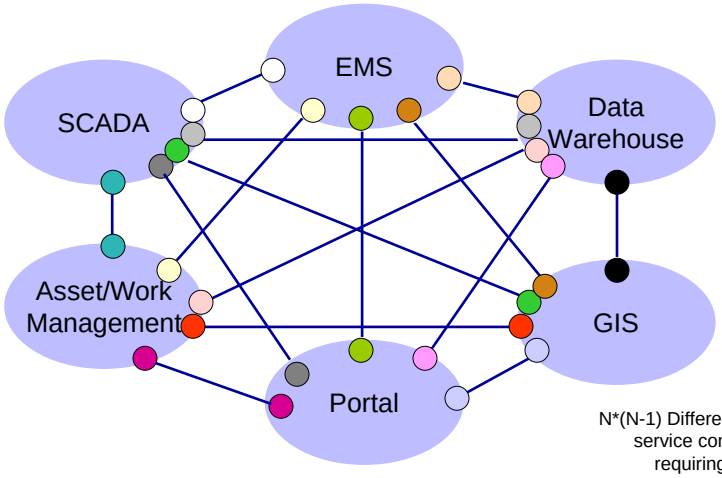
Basic Web Services Architecture



Service-Oriented Architecture (SOA)

- A ***service-oriented architecture (SOA)*** is an application framework that takes applications and breaks them down into individual business functions and processes, called ***services***.
- Services are the building blocks of an SOA.
  - A service can be a business function, such as find the location of an asset, or determine a voltage level at a certain time, or a system capability such as authenticate user.
  - SOAs enable “packaging” business functions from new and existing applications in a simple and standardized way.
  - An SOA increases flexibility by treating elements of business processes and the underlying IT infrastructure as components (services) that can be reused and combined to address changing business priorities.

Basic Web Service Integration Architecture



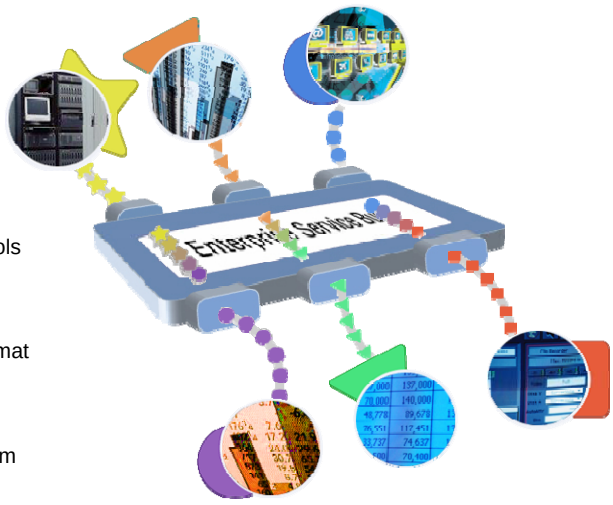
N*(N-1) Different client/server service combinations requiring coding

Typically no agreement on common messages, models, services, etc.

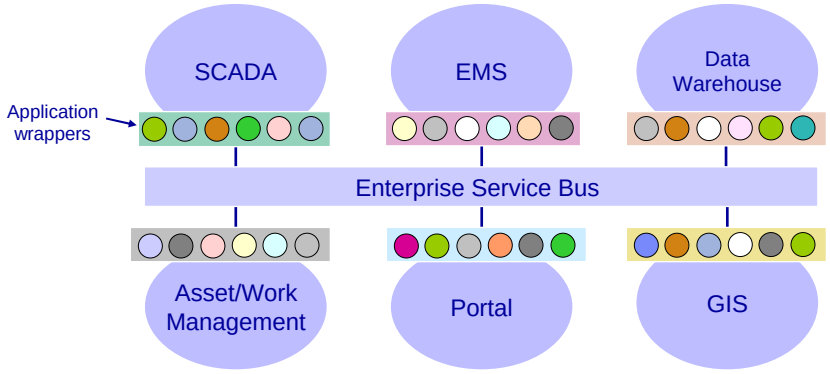
What is an Enterprise Service Bus (ESB)?

Flexible connectivity infrastructure for integrating applications and services to power SOA

- **ROUTING** messages between services
- **CONVERTING** transport protocols between requestor and service
- **TRANSFORMING** message format between requestor and service
- **HANDLING** business events from disparate sources



Components Connect To An ESB Using Web Services



SOA using ad-hoc web services with an ESB

- Significant benefits due to use of widely deployed technology:
  - SOAP – HTTP for transport
  - WSDL to describe the services and messages supported
  - Numerous development and middleware tools and products
- Ad-hoc because each application specifies its own services
  - Each service provider independently defines their own **SPECIFIC** web services
    - Get Customer Record
    - Create Customer Record, etc.
  - Each client needing to access a service must discover and adapt to each unique web service provider.
  - Application integration still requires significant programming effort unique to:
    - The specific application functions involved
    - The developer/brand of the applications
- Result: **integration is customized and unique to each and every system**

CIMug Meeting – 10 June 2008
Västerås, Sweden
GID Tutorial

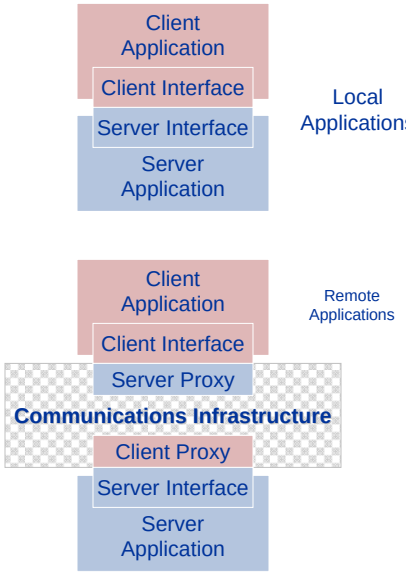
What About Interface Standards?

SISCO

SISCO

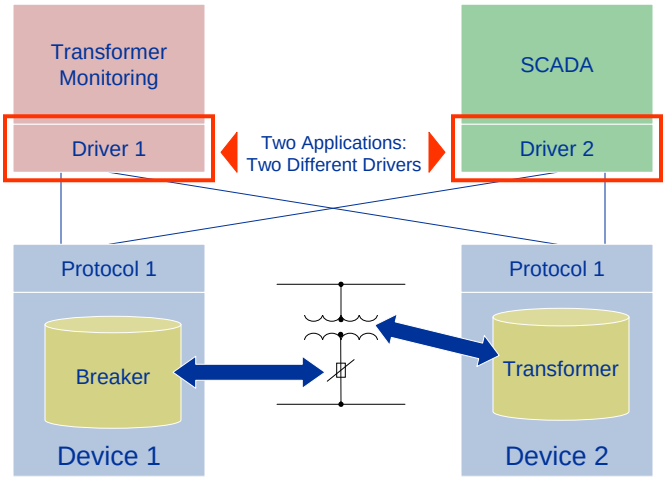
An Interface Standard

- Defines a standard for the interface between applications:
  - Independent of the means of communication (if any)
  - Independent of the specific application (generic)
  - Independent of the data exchanged (generic)
  - Independent of the developer (open)



CIM

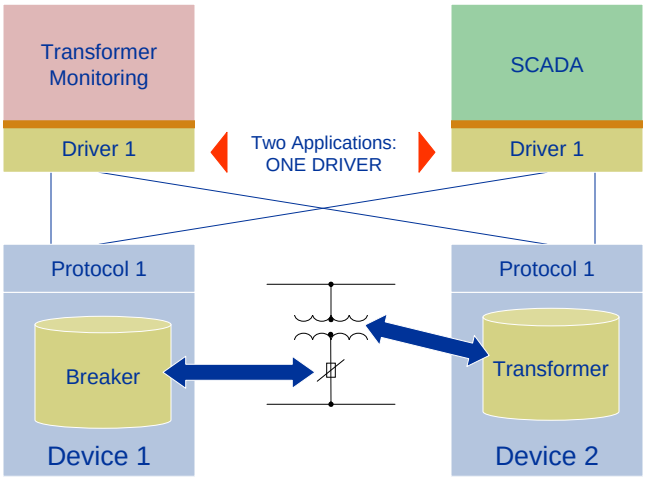
Genesis of the Interface Standard



Lack of Interface Standard

- Each application developer had to develop interfaces for any protocol that a user needs supported.
- Application developers spend considerable resources on drivers instead of applications.
- Fewer users per driver increases maintenance cost per user
- Result:
  - Less functionality
  - Higher costs
  - Proprietary solutions

Genesis of the Interface Standard



Impact of Interface Standards

- Allows developers to focus on applications because client application development can occur independent of the server application
- Enables sharing of interface development and maintenance costs across a larger user base
- Enables 3rd party development of add-ons, plug-ins, etc.
  - Enables niche application players to exist
- Results
  - More Functionality
  - More Choice
  - Less Proprietary
  - **Lower Costs**

Existing Widely Used Interface Standard: OPC

- OPC Foundation (http://www.opcfoundation.org) developed application programming interfaces to enable plug and play of applications and drivers called “OLE for Process Control” (OPC).
  - OPC Foundation: 362 member companies (end users and OEMs)
  - 1500+ companies developing OPC applications (est. OPC Foundation)
- OPC is dominant in the industrial automation and process control industries providing connectivity to **hundreds of key applications**.
  - 7500+ different OPC applications (est. OPC Foundation)
- Nearly **ALL** users in the industrial space expect and demand that their real-time application support OPC to simplify integration and **ALL** major application providers support OPC.

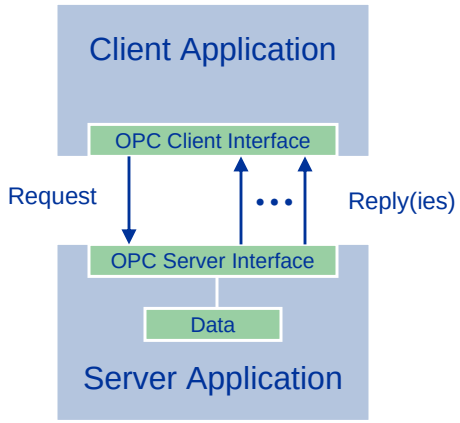
Some OPC Facts

- OPC is NOT a protocol – OPC is an INTERFACE
- Microsoft’s legacy approach to client/server distribution was DCOM
  - DCOM is not required for OPC
  - Alternatives to DCOM exist (including web services)
- Installing OPC does not automatically install DCOM or expose systems to DCOM vulnerabilities.
- Typical OPC installation is between client and server on the same computer with NO network between them.

OPC Interfaces

- Data Access (DA)
  - Client/Server Interface for reading, writing, and reporting real-time data
- Historical Data Access (HDA)
  - Client/Server Interface for accessing data archives
- Alarms & Events (AE)
  - Publish/Subscribe Interface for alarms and events

OPC Client/Server Architecture

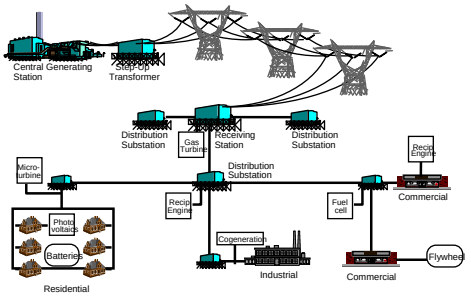


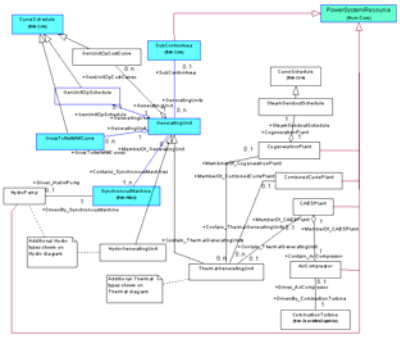
OPC Features

- Existing OPC Features
  - Based on Microsoft COM technology
  - C++, C#, Visual Basic, Java (3rd Party), and 1 WS Bindings
  - OPC Servers expose a namespace determined by the OPC Server itself
- Existing OPC Shortcomings
  - Microsoft specific
  - OPC Servers expose a namespace determined by the OPC server itself
    - Applications must adapt to each unique namespace presented by each OPC server depending on interface developer, device, application, protocol, etc.

OPC NEEDS A STANDARDIZED & TECHNOLOGY NEUTRAL METHOD OF REPRESENTING DATA

Common Information Model (CIM) is an object-oriented information model of the power system

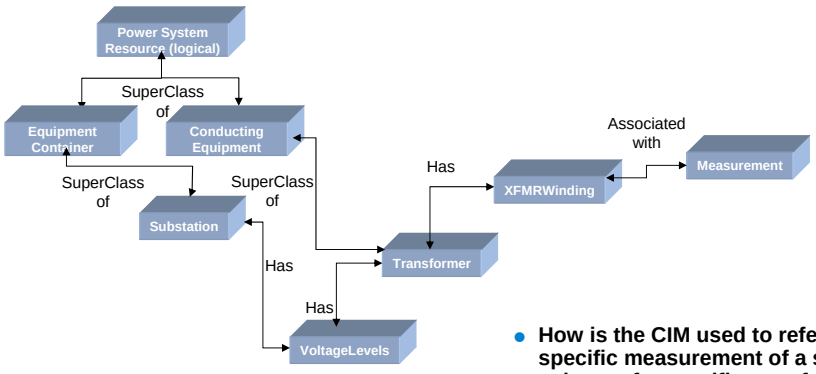




Class v.s. Instance

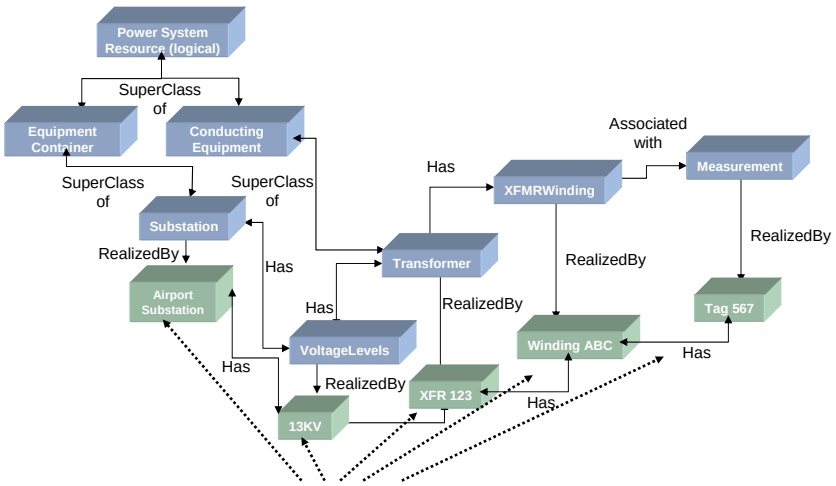
- Class information describes the objects and their properties and the relationships with other objects (e.g transformers are contained within substations, transformers have names, voltages, ratings, etc.).
- Instance information describes the specific objects of a given class that exist within the system.

The Common Information Model Defines Objects and Relationships



- **How is the CIM used to reference a specific measurement of a specific voltage of a specific transformer in a specific substation when there are many thousands of such voltages and they change every second?**

The Common Information Model As An "Instance Model"



Instances of Specific Power System Components

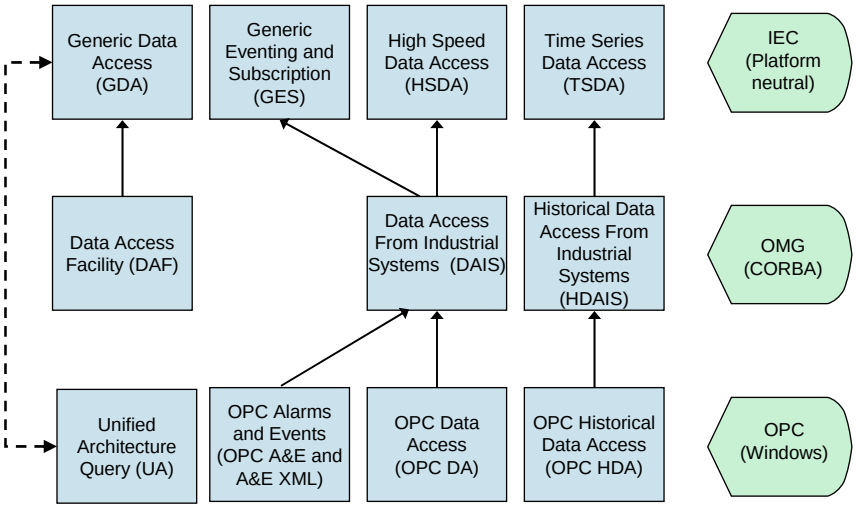
IEC 61970 and IEC 61968: Standards for Integration

- Provide a common agreement on What data is exchanged
  - The Common Information Model (CIM)
- Provide a common agreement on How to exchange the data
  - The Generic Interface Definition (GID)
- GID is a platform independent version of interface standards supporting model driven services capable of being used on any computing platform.
- EPRI CCAPI project submitted the Generic Interface Definition (GID) to the IEC WG13 for standardization in IEC 61970.

GID provides standardized interface services

- GID interfaces reference data in the context of a common data exchange model, the CIM.
- GID Services:
  - Generic Data Access (GDA): For model management and distribution of updates.
  - High-Speed Data Access (HSDA): For access to real-time measurement data.
  - Time Series Data Access (TSDA): For access to historical measurement data.
  - Generic Events and Subscriptions (GES): For pub/sub of generic XML messages.
- GID is based on existing open standards for both energy and industrial automation markets

GID Service Names And Lineage



GID: How data is exchanged

GID and OPC

GID Service	OPC
HSDA High Speed Data Access	DA Data Access
TSDA Time Series Data Access	HDA Historical Data Access
GDA Generic Data Access	UA Query Unified Architecture
GES Generic Eventing and Subscriptions	AE Alarms & Events

- GID is platform independent AND model-driven:
  - OPC uses vendor determined namespace for data
  - GID uses a namespace derived from the CIM for all data

GID Tutorial – Västerås

GID Standards

- IEC 61970 – 4XX
  - A Platform Independent Model (PIM) that describes with text and the Unified Modeling Language (UML)) the functionality of the GID
- IEC 61970 – 5XX
  - A Platform Specific Models (PSM) that applies the 61970-4xx series of specifications to particular platforms

GID Standard – IEC 61970-4xx

Part -4XX	Description
401	Component Interface Specification (CIS): Functional requirements and use cases for information exchange interfaces
402	Common Services: Base functionality of PIM
403	Generic Data Access (GDA)
404	High Speed Data Access (HSDA)
405	Generic Eventing and Subscription (GES)
407	Time Series Data Access (TSDA)

GID Standard – IEC 61970-5xx – Platform Specific Models

Part -5XX- Y	Description
502-7	C language Profile for Common Services (-402)
503-7	C language profile for GDA (-403)
503-8	Web services profile for GDA (-403)
504-7	C Language Profile for HSDA (-404)
504-8	Web services profile for HSDA (-404)
505-7	C language profile for GES (-405)
505-8	Web services profile for GES (-405)
507-7	C language profile for TSDA (-407)
507-8	Web services profile for TSDA (-407)

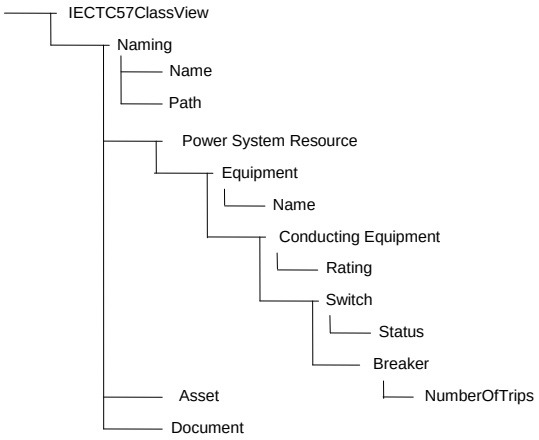
Common Services (-402): Identifiers

- Resource Identifiers (RID)
  - A 128-bit binary number that identifies a resource.
  - The numbering may have meaning defined by the GDA server although it may also be a Globally Unique Identifier (GUID).
- Uniform Resource Identifiers (URI)
  - A name that identifies a resource per IETF standard RFC2396.
    - mailto:ralph@sisconet.com - mailto scheme for the speaker
    - http://rfc.net/rfc2396.html - http scheme for RFC2396
    - http://utility.com/Planning/Production/CIM-schema-cimu09a#Transformer – definition of a transformer class object in a CIM model
  - A URI can change as the location of the object changes while the RID/GUID would not change as long as the object exists.

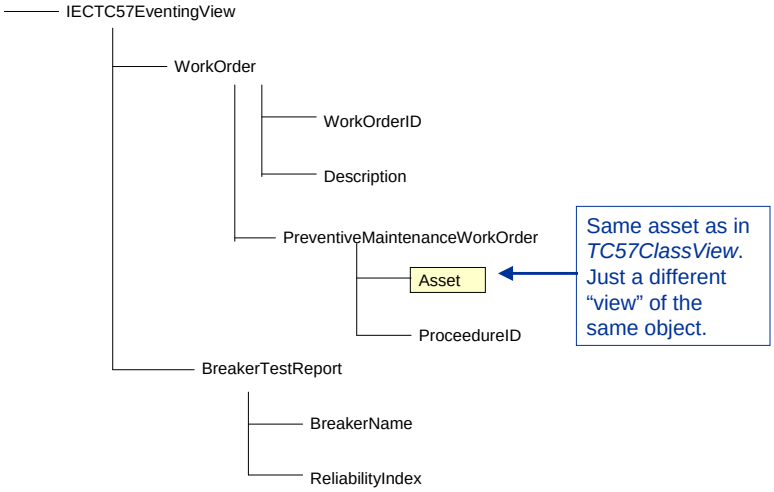
Common Services (-402): Views

- View is a method of representing a hierarchy determined by the model (CIM)
  - Can be used to derive a URI namespace
- GID supports 3 views:
  - TC57ClassView
  - TC57EventingView
  - TC57Physical View

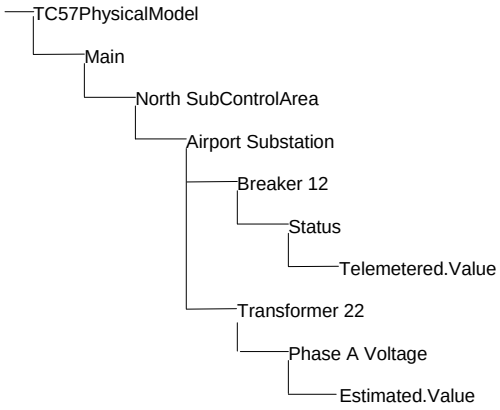
Example of TC57ClassView



Example of TC57EventingView

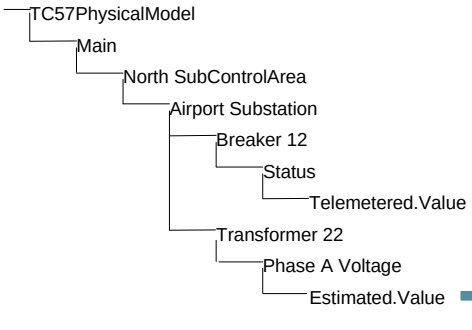


Example of TC57PhysicalView



TC57PhysicalView Namespace Example

- A namespace is created by collapsing a hierarchical view into a string that can be used as a Universal Resource Identifier (URI) to an object within that view.
- For Example:



utility.com/TC57Physical/Main/North/Airport/Transformer22/PhaseAVoltage/Estimated.Value

Generic Data Access (**GDA**), Part -403

- GDA provides generic request/reply services that are typically used to expose the model of a server (CIM) to clients.
- Typically used to access the model information (class and instance info) of a server that contains the common data exchange model (CIM) used by the system.
- GDA is a query interface that is independent of how data is stored
  - For instance, SQL is a query language that requires knowledge of table and column names, etc.
- Designed for an object oriented system such as CIM

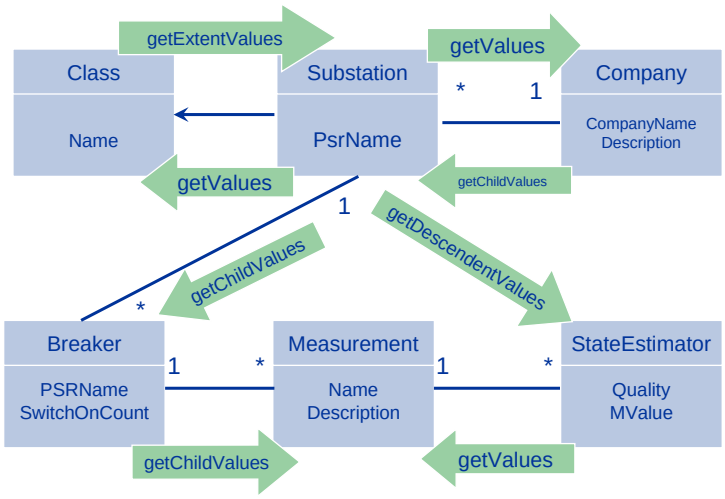
GDA Features

- GDA services enable clients to:
  - Read the data model (complete or partial)
  - Determine the hierarchy, relationships, associations, and attributes for the objects contained in the data model
  - Find objects in the data model via query
  - Interactively browse the model hierarchy.
  - Write/Create objects in the data model
  - Receive notifications when updates to the model are made.

GDA Resource Query Services

- get_values
  - Get a description of a resource specified by its ID.
- get_extent_values
  - Get a description of each resource for a given class
- get_related_values
  - Get description of resources associated to a given resource
- get_decendent_values
  - Generic form of a query encompassing the above queries that can be optimized for a specific type of query.

GDA Resource Query Services



GDA Advanced Queries

- Filtered Queries extend the Resource Query interface with the ability to specify property values that are used to qualify queries with operators like:
  - AND
  - OR
  - EQUAL
  - LESS/GREATER THAN
  - LIKE
- Extended Queries enable joining of data across multiple paths
  - For example, a client can ask for query data about breakers in a substation and for data about the substation itself in a single query

GDA Updates

- Enables a client to:
  - Create, delete, and update resources (objects) and their properties
  - Create, delete, and update object class definitions
- Supports concurrency and difference models
- Services:
  - create_resource_ids
  - create_resource
  - set_values
  - delete_resource
  - apply_updates

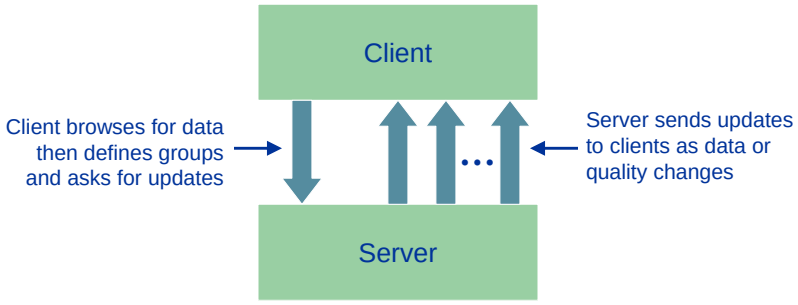
GDA Events

- Enables clients to be notified when changes to the model are done to a GDA server.
- Clients are informed of:
  - Resource IDs that have changed
  - Verbs defining the kind of change that occurred (deleted, created, etc.)
  - Version number to identify each unique change that occurred
- Clients use this information:
  - To update their internal configuration minimizing maintenance activities as the system evolves.
  - Can be used to drive additional queries to obtain detailed information to automate reconfiguration.

High Speed Data Access (**HSDA**) – Part 404

- Platform independent model (PIM) of the Platform specific model (PSM) of OPC Data Access (DA) for Microsoft Windows and Data Access for Industrial Systems (DAIS) for CORBA that was done by the Object Management Group (OMG – http://www.omg.org)
- Provides access to real-time data (current value) including quality and timestamp
- Uses a variety of services for data access:
  - Client browsing of data defined in server to use for access
  - Synchronous and asynchronous reads and writes
  - Group based asynchronous updates
    - Client defines groups that it is interested in
    - Server sends updates to the group to the client in an unsolicited manner

HSDA Group Update



HSDA Browsing Functions

- Enables a client to browse the data model of the server to find “items” (data comprising value, quality and timestamp)
- Browse attributes mapped to CIM with item names returned as URIs:

  utility.com/TC57Physical/Main/North/Airport/Transformer22/Phase%20A%20Voltage/Estimated.Value
- Handles returned are RID/GUID
  - Used in read, write, and update services to identify data
- Example Services
  - node_home()
  - item_home()
  - find_by_parent()
  - find_by_type()
  - get_pathname()
  - get_ids()

Time Series Data Access (**TSDA**) – Part 407

- Platform independent model (PIM) of OPC Historical Data Access (HDA) for Microsoft Windows and Historical Data Access for Industrial Systems (HDAIS) for CORBA that was done by the Object Management Group (OMG – http://www.omg.org)
- Provides access to data, including quality and timestamp, based on the item description and the range of time for which the data is requested
- Used for historical data access
- Similar services and usage as HSDA

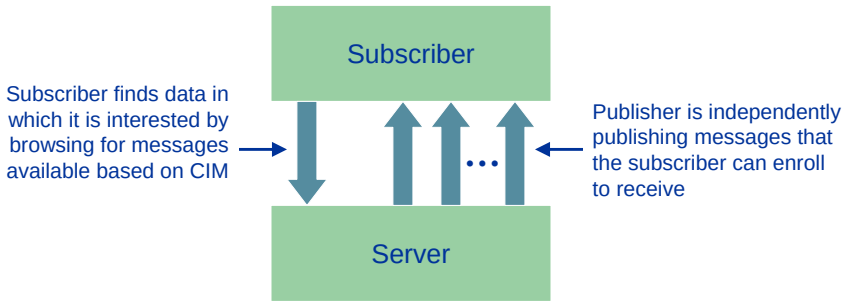
Generic Eventing & Subscriptions (**GES**) – Part 405

- Interface for publishing and subscribing to generic XML messages
- Does not specify message payload
  - Determined by application/system or standard (IEC 61968)
- Supplies utility specific features
  - Model driven publish/subscribe to generic XML messages using object references in the CIM context
- Middleware independent pub/sub interface enables application portability across platforms
- Based on OPC Alarms & Events (AE) using HSDA like browsing features

GES Services

- Superset of OPC alarms and events
  - Alarms may be CIM elements
  - Events consist of CIM objects (e.g. Work order)
- Example services
  - get_filter ()
  - set_filter ()
  - get_state()
  - set_state ()
  - query_available_filters ()
  - set_subscription ()

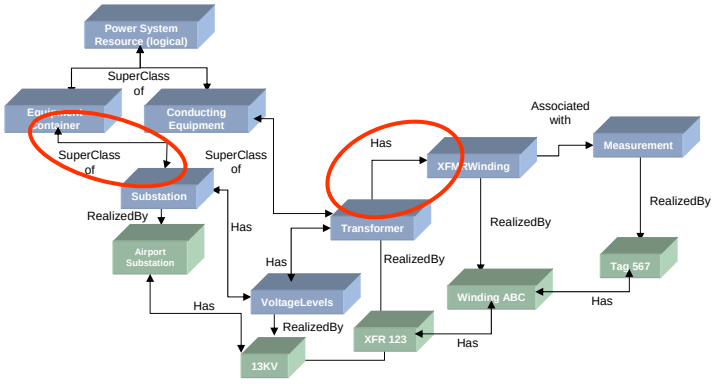
GES



CIMug Meeting – 10 June 2008
Västerås, Sweden
GID Tutorial

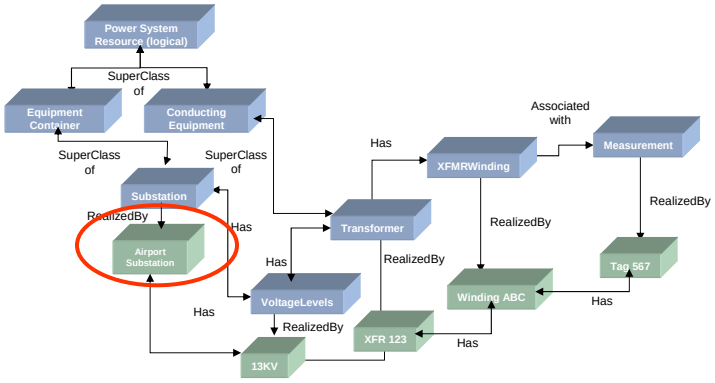
GID Example Usage

Example of GDA Application



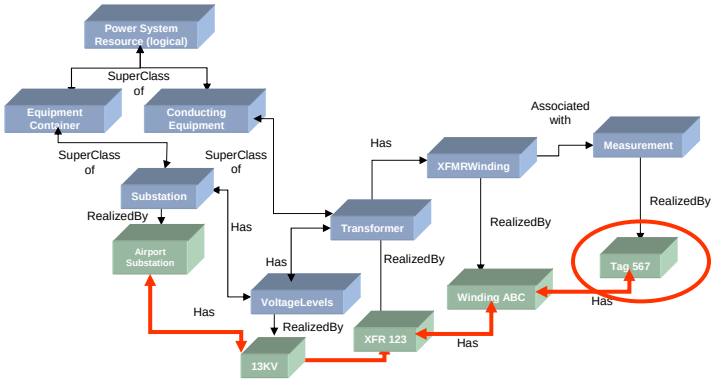
- Discover the relationships and associations between CIM classes

Example of GDA Application



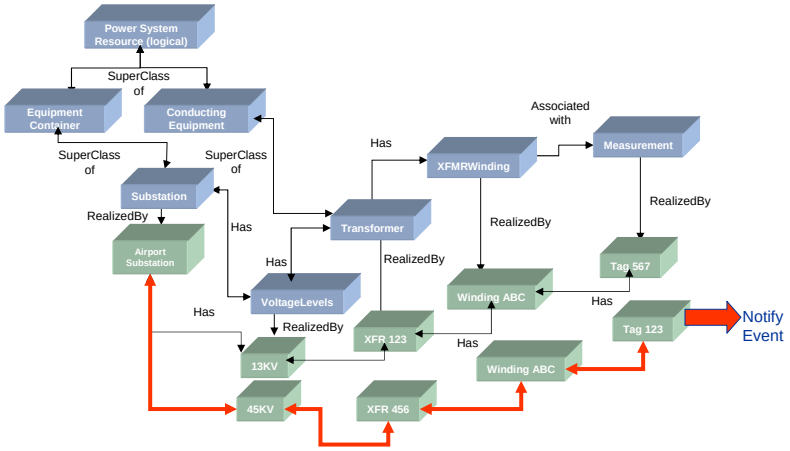
- Query for a list of power systems resources of class “substation”

Example of GDA Application



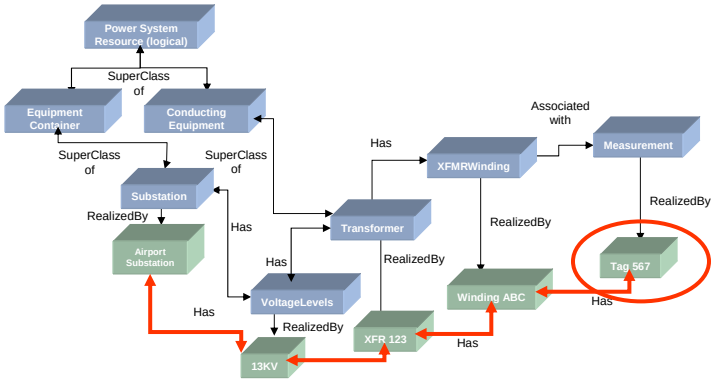
- Query connectivity and containment to find specific transformers in specific substations and find the identifier for a specific measurement.

Example of GDA Application



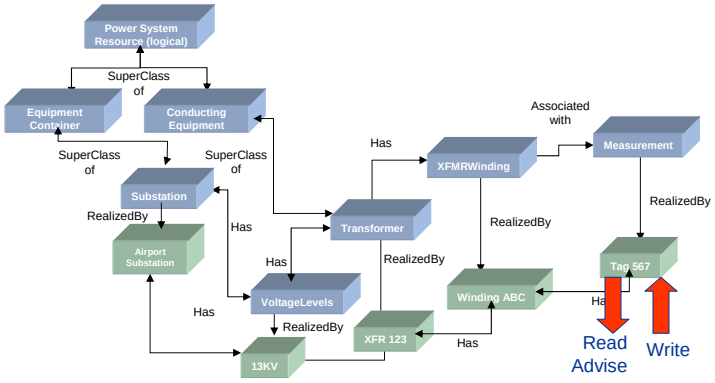
- Create (or delete) new Instances in the model and notify others of their existence.

Example of HSDA Application



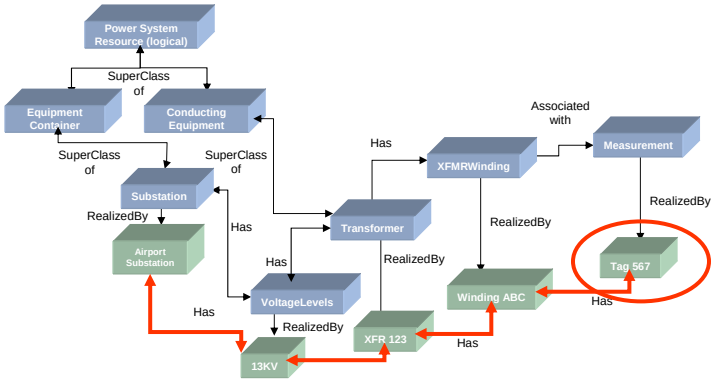
- Browse a hierarchical “view” of the model to find specific measurement values to get an identifier.

Example of HSDA Application



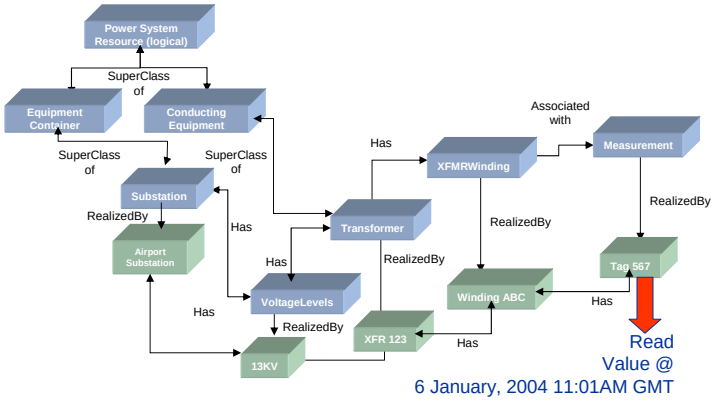
- Once the identifier is known, ask to receive updates to that measurement value

Example of TSDA Application



- Browse a hierarchical “view” of the model to find specific measurement values to get an identifier.

Example of TSDA Application



- Once the identifier is known, ask to read the value that existed at a specific point of time or a sequence of values over a period of time.

Why not use GDA for Browsing Measurements?

- You Can. The GDA service can support this.
- Typically, the model server does not contain the measurements.
- Typically, the system with the measurements does not expose the model.
- Use the appropriate GID service for the Job

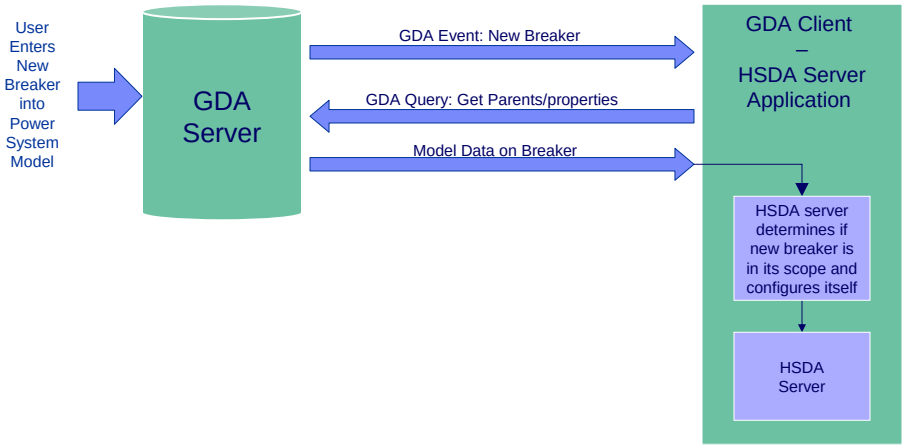
Why not use HSDA/TSDA Browsing to get Model Info?

- You can, BUT:
  - HSDA/TSDA Browsing only traverses a hierarchical view of the full-mesh CIM model.
  - Does not enable discovery of all possible relationships.
    - Only the relationships that exist in the chosen view are exposed.
  - Does not enable distribution of model update notices or updating or modifying the model.
- Use the appropriate GID service for the Job

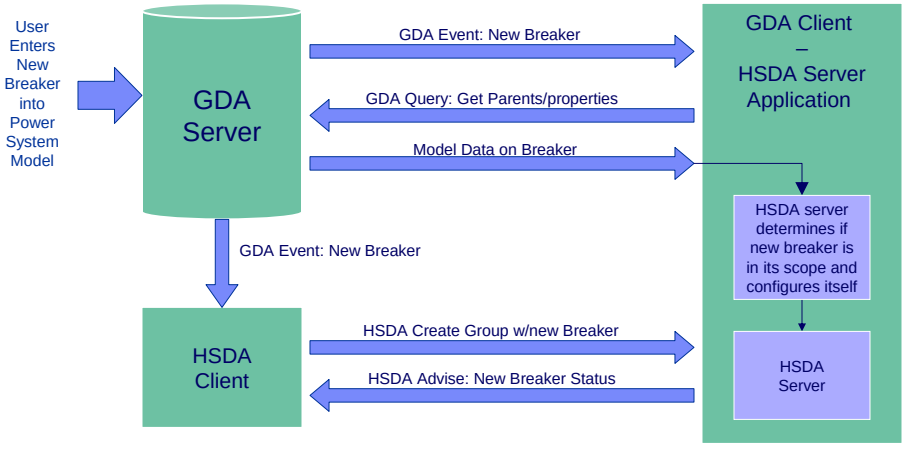
Some GID Applications

- Enable model information to be shared and exposed over a message bus to provide a unified data exchange model for applications.
  - Eliminate application dependencies on internal data representations of systems.
  - Enable self-configuring clients.
  - Simplify propagation of system changes across enterprise.
- Expose models and data from legacy systems as CIM data
  - Eliminates dependencies on legacy table formats from applications needin data
  - Enables lower cost application migrations and ehnancements
  - Enables data trapped in inflexible legacy systems to be exposed and leveraged without copying/reproduction of the data

GDA Application



GDA Application



CIMug Meeting – 10 June 2008
Västerås, Sweden
GID Tutorial

Application of GID to Web Services

Web Service Architecture

- Web service tools and development environments are very good at maximizing the productivity of developing and consuming web services
- But, without agreement and detailed standards on generic services that all applications can use, integration is ad-hoc and unique to each system
- GID can be translated into WDSL to create standardized and generic web services that are:
  - Developer independent
  - Application function independent
  - Middleware independent
  - User independent
- Can enable off-the-shelf reusable application adapters from third parties

GID Web Services

Part -5XX- Y	Description
502-7	C language Profile for Common Services (-402)
503-7	C language profile for GDA (-403)
503-8	Web services profile for GDA (-404)
504-7	C Language Profile for HSDA (-404)
504-8	Web services profile for HSDA (-404)
505-7	C language profile for GES (-405)
505-8	Web services profile for GES (-405)
507-7	C language profile for TSDA (-407)
507-8	Web services profile for TSDA (-407)

HSDA in C language OPC PSM (-504-7)

```
typedef struct {
        FILETIME            ftStartTime;
        FILETIME            ftCurrentTime;
        FILETIME            ftLastUpdateTime;
        OPCSERVERSTATE      dwServerState;
        DWORD               dwGroupCount;
        DWORD               dwBandWidth;
        WORD                wMajorVersion;
        WORD                wMinorVersion;
        WORD                wBuildNumber;
        WORD                wReserved;
        [string] LPWSTR     szVendorInfo;
} OPCSERVERSTATUS;
```

```
typedef enum tagOPCSERVERSTATE {
    OPC_STATUS_RUNNING = 1,
    OPC_STATUS_FAILED,
    OPC_STATUS_NOCONFIG,
    OPC_STATUS_SUSPENDED,
    OPC_STATUS_TEST } OPCSERVERSTATE;
```

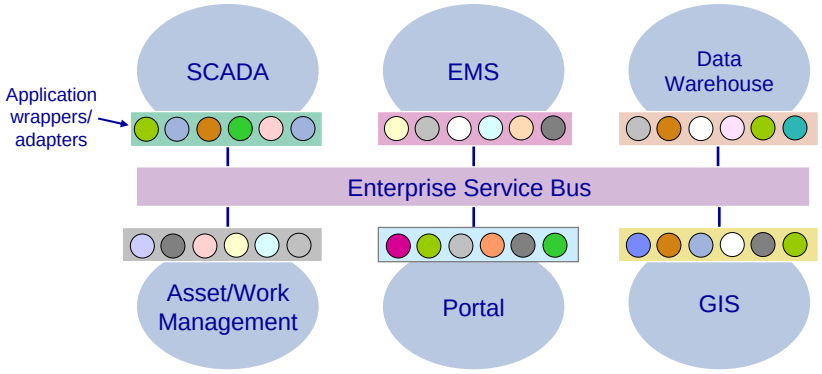
HSDA in WSDL for Web Services PSM (-504-8)

```
- <s:simpleType name="serverState">
  - <s:restriction base="s:string">
      <s:enumeration value="running" />
      <s:enumeration value="failed" />
      <s:enumeration value="noConfig" />
      <s:enumeration value="suspended" />
      <s:enumeration value="test" />
      <s:enumeration value="commFault" />
    </s:restriction>
  </s:simpleType>
- <s:complexType name="ServerStatus">
  - <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="StatusInfo" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="VendorInfo" type="s:string" />
      <s:element minOccurs="0" maxOccurs="unbounded" name="SupportedLocaleIDs" type="s:string" />
      <s:element minOccurs="0" maxOccurs="unbounded" name="SupportedInterfaceVersions"
        type="s0:interfaceVersion" />
    </s:sequence>
    <s:attribute name="StartTime" type="s:dateTime" use="required" />
    <s:attribute name="ProductVersion" type="s:string" />
  </s:complexType>
```

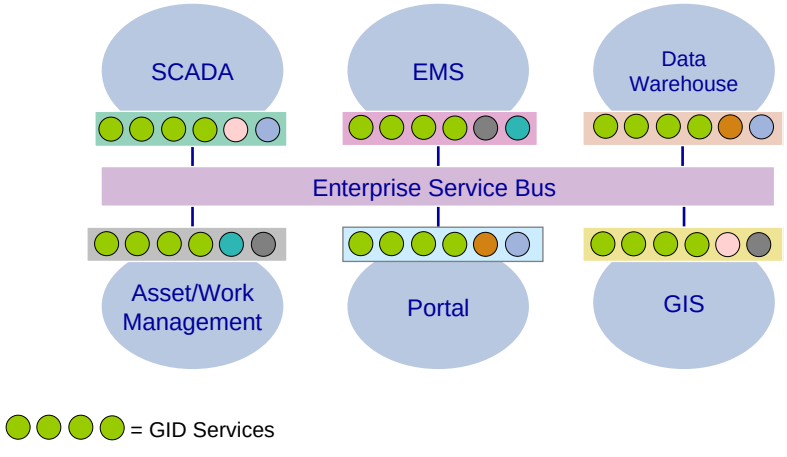

Web Service Architecture

- Web service tools and development environments are very good at maximizing the productivity of developing web services based apps.
- But, without agreement and standards on generic services that all applications can use, integration is ad-hoc and unique to each system
- GID can be used to create standardized and GENERIC web services that are:
  - Developer independent
  - Application function independent
  - Middleware independent
  - User independent
- Can enable off-the-shelf reusable application adapters from third parties

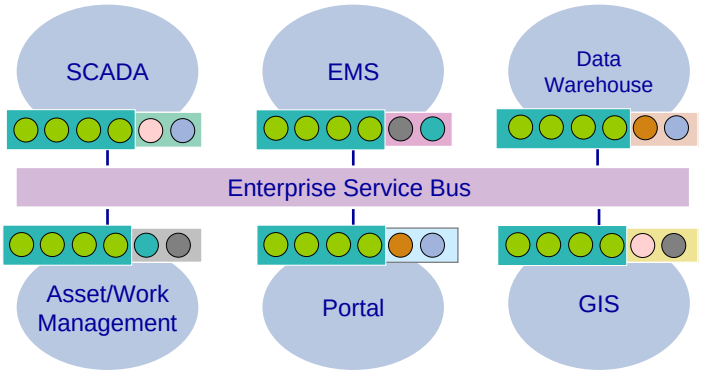
Ad-Hoc Integration Using Web Services



GID Simplifies Integration using Common Generic Services

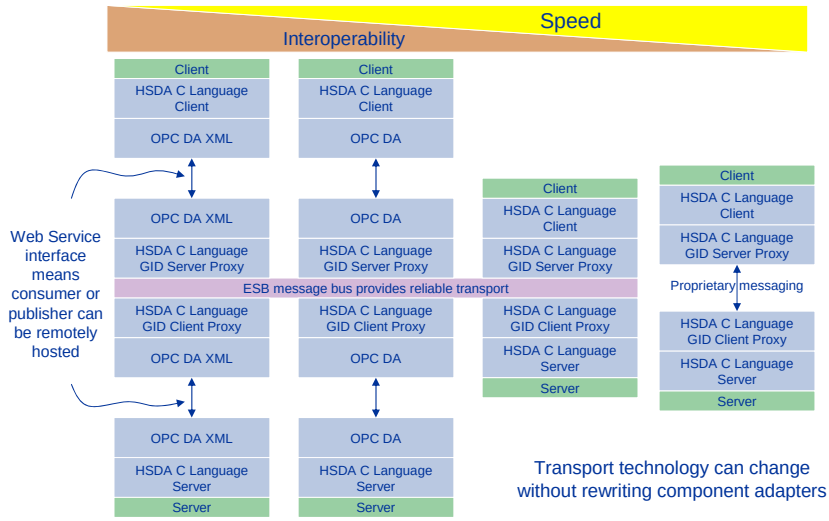


GID Enables Real Plug 'n' Play for Integration



= Reusable Productized Adapters

Use GID as Appropriate



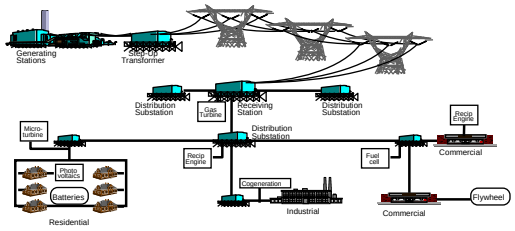
CIMug Meeting – 10 June 2008
Västerås, Sweden
GID Tutorial

Why Isn't Meta-Data Modeling Enough for the Power System Operations of Utilities?

What makes utilities so special?

GID Tutorial – Västerås

Small Municipally Owned Utility



- 370 Miles of transmission wiring (hundreds of towers, thousands of insulators and connections)
- 3,600 miles of distribution wiring (many thousands of towers, tens of thousands of insulators, connections)
- 7 transmission substations with several hundred separate assets including devices, transformers, switches, insulators, breakers, controls, capacitor banks, etc. Each asset can have hundreds (sometimes thousands) of real-time measurements and hundreds of attributes to describe the asset.
- 40 distribution substations with 1,300 distribution devices (see above).
- Several generation stations each with its own transformers, breakers, and process controls.

82

Scaled Up: A Medium to Large Utility

- Hundreds of transmission substations with thousands of miles of wire and thousands of towers
- Thousands of distribution substations with tens of thousands of miles of wire and many tens of thousands of poles, transformers, breakers, etc.
- Numerous Generation stations
- Tens of thousands of devices
- Many hundreds of thousands of measurements possible and **desired**
- Add Metering to this mix: **Millions** of measurements every few minutes
- All of this information is **CRITICAL** to the business

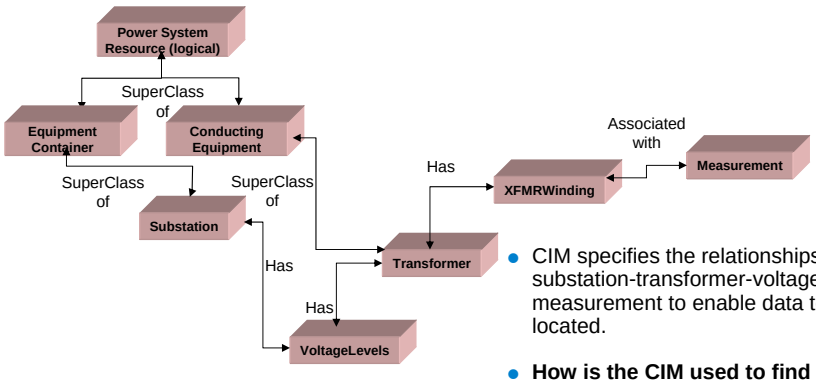
Business Applications

- A typical business application uses objects like:
  - Purchase Orders
  - Assets
  - Customers
  - Invoices
  - Payments
  - Etc.
- There are a large number of objects whose structure changes infrequently.
- There are numerous transactions and business interactions (processes) that must be managed by the integration infrastructure

Operational Applications

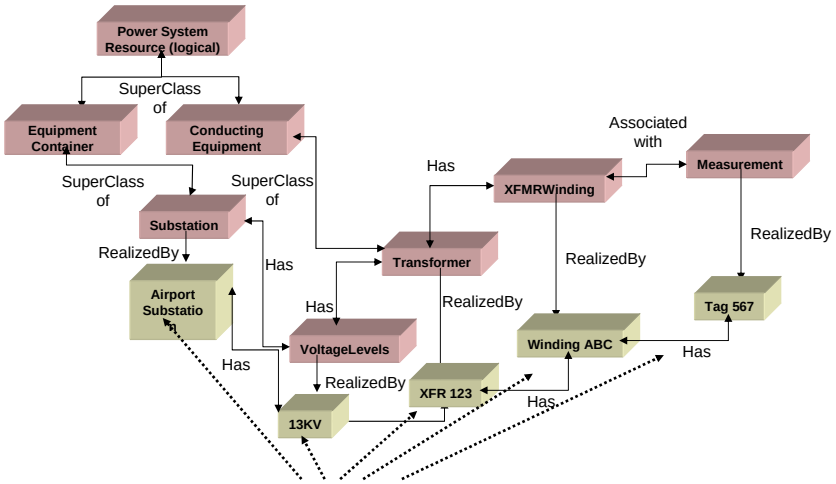
- Typical operational application will have objects like:
  - Feeders
  - Breakers
  - Transformers
  - Lines
  - Switches
- There are a large number of these objects and their attribute values, in many cases, **change constantly (e.g. once per second).**
- Many attributes like voltages, currents, and operations are monitored constantly resulting in many thousands (millions) of transactions (data value updates) every day for many of these objects.

The Common Information Model Defines Objects and Relationships



- CIM specifies the relationships between substation-transformer-voltagelevel-measurement to enable data to be located.
- **How is the CIM used to find a specific measurement of a specific voltage of a specific transformer in a specific substation when there are many thousands of such voltages and they change every second?**

The Common Information Model As An "Instance Model"



The Impact of Instance Models on Integration Technology

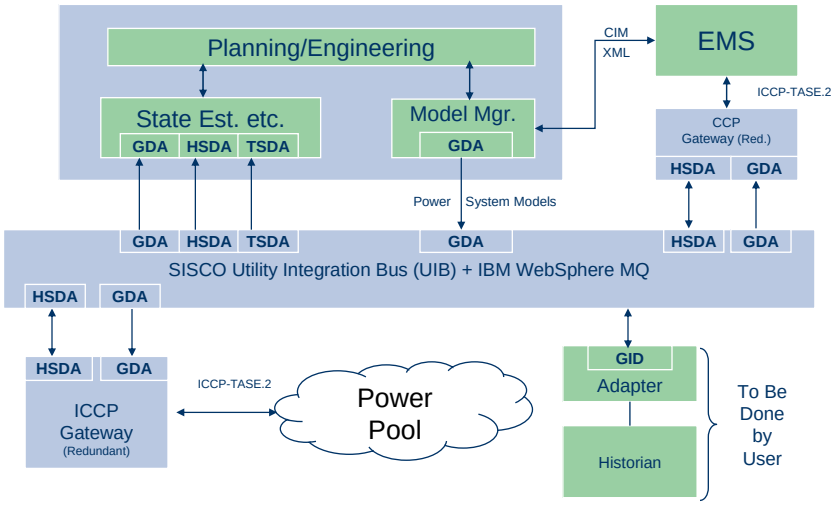
- The normal object mapping and transformation tools of an ESB that work so well for transaction oriented business systems are not optimized for this.
- Efficient instance mapping at the application adapter is needed.
  - The instance mapping only needs to be done by applications that serve data to other client applications.
- Client applications find the name they need by invoking GID services with a model server that has instance knowledge.
- Client applications then invoke GID services to obtain the data from the application adapter (server).
- Client applications don't need to be configured with knowledge of the instance mapping in the servers and don't have to be changed as the server applications change.

CIMug Meeting – 10 June 2008
Västerås, Sweden
GID Tutorial

Example

GID Tutorial – Västerås

Sample Project 1

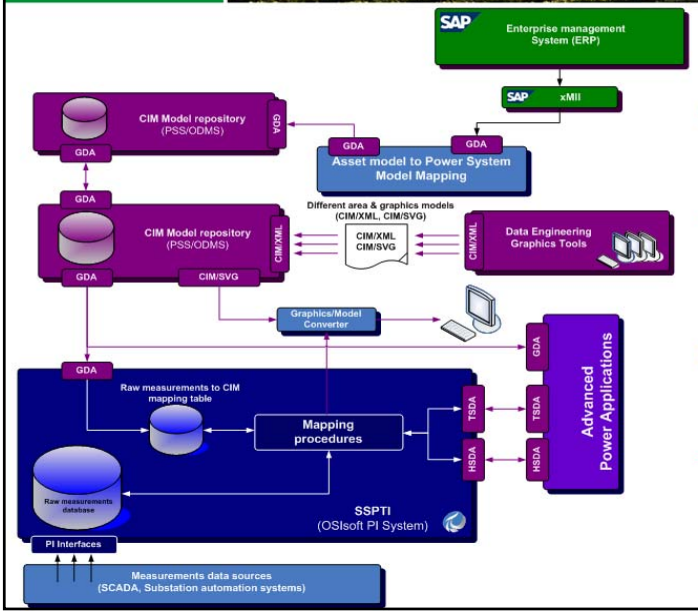


90

System Overview

- Model Management Separated with Exposed GID Interface
  - Enables sharing of models across functional boundaries.
    - Weeks to days to get extensions operational.
  - Enables use of models for external integration.
  - Isolates changes in data base formats from other applications.
- Use of GID services for model and data
  - Enables model data to be used in configuration of the data exchange links.
  - Model changes can be detected and ICCP links configured semi-automatically.
  - Enables off-the-shelf products from other companies to plug-in to the environment using off-the-shelf adapters.
  - Enables user development and expansion.

ASTU target architecture



- Asset and Power System Resource models linking
- Distributed model synchronization between multiple CIM repositories
- Graphics models exchange with CIM objects attributes
- Combined usage of Power System Resource model and measurements data

Additional Example

- Thursday: 9 am
  - Application of Model Driven Integration on Oil Platforms - Jone Aarre, IBM

CIMug Meeting – 10 June 2008
Västerås, Sweden
GID Tutorial

Conclusion

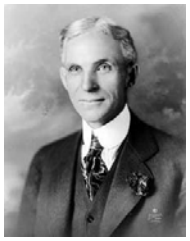
GID Enables Independent Development

- An open interface standard enables development and support of application adapters independent of the applications and independent of a specific user data (a GENERIC approach):
  - Makes off-the-shelf application adapters viable which can reduce or even eliminate the need for application adapter development.
  - Enables 3rd parties to provide niche off-the-shelf interface products that application vendors are either uninterested or too expensive.
  - Generic adapters using generic services can be used cross industry increasing product availability and competition lowering costs.

Do Interface Standards Work?

- The use of generic interface standards is widely used, accepted, and demanded in the industrial automation industry based on the OPC Foundation Standards.
  - 300+ members of vendors and users
  - 1500+ plus companies supporting OPC products
  - 7500+ plus products available
  - Tens of Thousands of installations in mission critical systems
- OPC Unified Architecture (UA) is a secure web service based version of the OPC MS-COM based interfaces with hundreds of companies implementing.
- IEC 61970 is expected to include new PSMs for GID referencing OPC UA mappings.

There is no reason that the energy industry can't take advantage of standardized interface services to lower integration costs.



"Obstacles are those frightful things that appear when you take your eyes off your objective."

- Henry Ford

- Ralph's Corollaries:
  - If you don't set out on the path, you will never make any progress.
  - The first on the path will reap the most reward.

Questions - Discussion

CIMug Meeting – 10 June 2008
Västerås, Sweden
GID Tutorial

Thank You

Ralph Mackiewicz
SISCO, Inc.
6605 19½ Mile Road
Sterling Heights, MI 48314-1408 USA
Tel: +1-586-254-0020 x103
Fax: +1-586-254-0053
Email: ralph@sisconet.com