Research on Heterogeneous Data Integration for Smart Grid

Bin Lu
Department of Computer Science
North China Electric Power University
Baoding, 071003, Hebei Province, China
binlu714@yahoo.com.cn

Wei Song
Department of Computer Science
North China Electric Power University
Baoding, 071003, Hebei Province, China
huadiansongwei@163.com

***Abstract*—Compared with the traditional power grid, Smart Grid can improve the reliability and availability of the entire grid, while it continues to reduce the costs and imorove the efficiency. It is getting more and more attentions, the characteristics of integration and interaction require the exchanging and sharing of the information, as well as the seamless connection of various parts. The Ontology as a shared conceptual model and formal specification provides a good way to resolve the semantic heterogeneity problem. By the research of past means of heterogeneous data integration, combined with the cuerrent status of the grid, a model of heterogeneous data integration is proposed to adapt the future Smart Grid. Based on XML and Ontology, combined with Cloud services, This model provides a good method to solve the heterogeneous problem from the syntax and sematics. Finally, SCADA(Supervisory Control and Data Acquisition) data is used to valiadate the model.**

Keywords-SmartGrid, Ontology, SCADA, Heterogeneous data, Cloud services;

I. INTRODUCTION

The continuous development of information society made a higher demand on the electricity market. The coordination between electricity systems, electricity market and users is getting closer, more frequent exchange of information, and increasing number of distributed generation resources, such as renewable energy, digital power grid has become the trend.

The concept of Smart Grid was first put forward in2003. In accordance with the IBM，the “Smart Grid” has three characteristics[1]: first, a higher degree of digitalization, containing a variety of intelligent sensors, such as electrical equipment, control systems, application systems, etc, and connecting more devices; second, it is based on a unified information platform, which can finish the integration of data and application automately; third, it is based on business intelligence analysis system, and has the capabilities of assisting decision support for data analysis, that is, to optimize the operation and management according the correlation analysis of existing power grid data. Unlike the past, data integration from a sectoral point of view, "Smart Grid" from the perspective of data integration of assets.

978-1-4244-5540-9/10/$26.00 ©2010 IEEE

II. BASIC CONCEPTS

A. XML and Ontology

XML and HTML are all from SGML, but XML is easier to understand, as the next-generation language, it has neihter a label, and no syntax.

Several characteristics of XML document are as following[3]: the first is its scalability, refers to that the user can freely define their own data tags; the second is structured, namely, data structure can model any complex things that have different levels; validity is the third, data can check the validity of the structure; what is more, XML has the characteristic of interoperability, and self-describing, you can use it in a variety of platforms; also it can be interpreted through a variety of tools and supports for many of the major criteria which used for characters encoding.

Ontology was first putforward as a philosophical category, is a systematic explanation, and used to describe the nature of things. Subsequently, with the development of artificial intelligence, a new definition is given to Ontology by artificial intelligence. That is, Ontology is a shared conceptual model which is clear and formal. Ontology has four meanings[4]: conceptual model, explicit, formal, shring.

B. Cloud services of IBM

IBM has put forward the following three types of cloud services[5]:

- Software as a Service (SaaS): this is the most common type of cloud services, most of us more or less have used. In this model, the service provider provides all the service. Users interact with this service through a Web-based front-end. The service covers a wide range, from Web-based e-mail, such as GMail, to financial software, such as Mint.
- Platform as a Service (PaaS): a service which provides software and product development tools, these tools are hosted on their hardware infrastructure by the provider. Users create applications by this platform as well as the API provided by it or a graphical user interface. Typical examples of such services are Salesforce.com, Force.com, and Google App Engine.
- Infrastructure as a Service (IaaS): a service provide -s accessing to the the basic building blocks. We can build infrastructure which is needed to run an application by combining and stratifing these blocks.

The most typical examples are Amazon Web Services (AWS)and Rackspace.

IaaS is used in this article, we deploy the three components of the the middle layer to the cloud platform of power system, if other applications need these features, they can directly access to the cloud platform to implement custom applications.

III. System Technology

System adapts the approach of combination of components and middleware, component technology is used to provide plug and play function, and the message-based middleware to enable heterogeneous integration. Component is a software code unit which encapsulates the data and logic by itself, as well as a reusable software component. Within the components, and also the combination of hand-written codes can be reused, in order to accept the user's custom.

Components can be called through the port, this requires the code which be packaged to provide a standard interface, Component model defines the structure of the interface and a variety of mechanisms of components and component containers. At present, three component technologies are popular, they are EJB, CORBA, and COM.

Smart Grid requires the system that has full integration capability between applications, this also requires components to provide some or all of the following features[6].

- Any complex information can be exchanged between components;
- To achieve this, different forms of distributed component technology be able to use;
- information can be reused;
- To meet the interaction between administrators and components;
- To provide the facilities of common event history as a component;
- To support the event historical patterns components which based on element model of information exchange;
- Provide a historical component to record the time which send by distributed components when releasing the event;
- Be able to support the event information model versions and component versions;
- Provide monitoring component between applications, in order to analyze the state of any component interface that connecting to the application specific services of power companies;
- Allow components to send or request information without need to know the physical location of receiveing components, or whether they are connected.

Middleware is used to describe a group of software products, which have the role of integration, conversion or translation. At the same time, middleware provides some common interface for the event, message, data access, services, etc.

Between system and application software is the middleware, which gives an abstract of the application software, combines the technical details of large number of applications, and provides the separation mechanism of application softwares and data sources.

IV. System Dedign

Currently in power systems there is no uniform data formats, it is difficult for data sharing and interaction, as well as other issues due to many types of data and information. Heterogeneous mainly shows in two aspects: system heterogeneous and model heterogeneous. There are many ways to solve the heterogeneous problem, the main also the common as follows: IBM's proprietary federated database systems (Federatedsyste, FDBS), Data Warehouse, Mediator, etc. The technology above is used to sovle integration and interoperability of the multiple data sources of distributed environment.

A. *Symtem Model*

XML has played a tremendous role in resolving syntax heterogeneous, however, there are still heterogeneous of semantics, the application of ontology technology can be a good solution to semantic heterogeneity. In this paper, the combination of XML and Ontology is adopted to solve the heterogeneous of syntax and semantics of power system integration, in order to build the unified information platform under Smart Grid. The model is based on three-tier architecture of B/S, namely, the application layer, middle layer and data layer, this structure gives full play to the B/S and C/S architecture advantages, has realized thin-client, distributed applications, and transparent access. System model is shown in Figure1.

In accordance with the actual situation in power system, the application layer includes two types of users: one is the administrator workstation which designed specially for the staff of power system. Managers monitor the real-time data comes from the scence and various systems through a dedicated software, then give the corresponding operations to ensure that all the systems running nomally; the other is for people who use the facilities for query, the most important terminal is the browser, followed by telephone, voice, and SMS, also mobile phone platform (wap), that will meet the needs of the future Smart Grid.

The middle layer is mainly responsible for the communication, integration and exchange of the heterogeneous data. This layer combinating middleware with component to achieve data integration, mainly consists of three main parts: ontology component, schema mapping component, and query component.

In the environment of the future Smart Grid, by establishing the cloud platform of the power system, we can achieve the aims of saving source of hardware and software, scheduling reasonable, and giving balance of the source, as well as co-ordinating the electricity market. Taking the security requirements of the power system into account, the above three components are regarded as "private cloud", respectively, the entire middleware as the "private cloud" of the power system. When other applications or customizations need some of the above, or the whole functions, then you can directly access the cloud platform to call this function in order to assemble and reuse, without re-developing, this implement reusability and Plug and Play functionality of component.

B. *System Implementation*

Ontology component is divided into the global ontology and local ontology. The database used by each application is different, as well as the development platform, and operation system, so it is necessary to establish the appropriate local ontology through the schema mapping for each application system, and then build the global ontology on the basis of local ontology for the entire electric power system.

Map the underlying source to corresponding local ontology by schema mapping and homologous wrapper. The following are parts data sheeets of telemetry (Telemetry), RTU, and PowerSystemResource.

Telemetry(Yccode, YCDescription, YCCZcode, YCRTU Ycnum, YCSafeL);

RTU(BaseAddress, Characteristics, ObjectAddress, RtuProtocol, PowerSystemResourceName, PowerSystem-ResourceDescription);

PowerSystemResource(PowerSystemResourceDescrition, TypeName, Pathname, LoadFile, PowerSystem-ResourceName).

The schema file of telemetry table is “Telemetry.xsd”, as following:

```
<?xml version="1.0" encoding="gb2312" ?>
<xsd:schema
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:IEC61970="http://www.iec.org/61970"
xmlns:cim="http://iec.ch/TC57/2000/CIM-schema-
cimu09b#"
xmlns:dt="urn:schemas-microsoft-com:datatypes">
  <ElementType name="YCcode" dt:type="Char" />
  <ElementType name="YCCZcode" dt:type="Char" />
  <ElementType name="YCDescription" dt:type="Char"
/>
  <ElementType name="YCRTU" dt:type="Smallint" />
  <ElementType name="YCnum" dt:type="Smallint" />
  <ElementType name="YCSafeL" dt:type="Tinyint" />
  <ElementType name="Telemetry" content="eltOnly">
  <element type="YCcode" />
  <element type="YCCZcode" />
  <element type="YCDescription" />
  <element type="YCRTU" />
  <element type="YCnum" />
  <element type="YCSafeL" />
  </ElementType>
  </xsd:schema>
```

The next is RTU.xsd :

```
<?xml version="1.0" encoding="gb2312" ?>
<xsd:schema
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:IEC61970="http://www.iec.org/61970"
xmlns:cim="http://iec.ch/TC57/2000/CIM-schema-
cimu09b#"
xmlns:dt="urn:schemas-microsoft-com:datatypes">
  <ElementType name="BaseAddress" dt:type="Char" />
  <ElementType name="Characteristics" dt:type="Char"
/>
  <ElementType name="ObjectAddress" dt:type="Char"
/>
  <ElementType name="RtuProtocol" dt:type="Char" />
  <ElementType name="PowerSystemResourceName"
dt:type="String" />
  <ElementType name="PowerSystemResource-
Description" dt:type="String" />
  <ElementType name="RTU" content="eltOnly">
  <element type="BaseAddress" />
  <element type="Characteristics" />
  <element type="ObjectAddress" />
  <element type="RtuProtocol" />
  <element type="PowerSystemResourceName" />
  <element type="PowerSystemResourceDescription" />
  </ElementType>
  </xsd:schema>
```

The PowerSystemResource.xsd as followed:

```
<?xml version="1.0" encoding="gb2312" ?>
<xsd:schema
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:IEC61970="http://www.iec.org/61970"
xmlns:cim="http://iec.ch/TC57/2000/CIM-schema-
cimu09b#"
xmlns:dt="urn:schemas-microsoft-com:datatypes">
  <ElementType name="PowerSystemResource-
Description" dt:type="Char" />
  <ElementType name="TypeName" dt:type="String" />
  <ElementType name="Pathname" dt:type="String" />
  <ElementType name="PowerSystemResourceName"
dt:type="String" />
  <ElementType name="LoadFile" dt:type="String" />
  <ElementType name="PowerSystemResource"
content="eltOnly">
  <element type="PowerSystemResourceDescription" />
  <element type="TypeName" />
  <element type="Pathname" />
  <element type="PowerSystemResourceName" />
  <element type="LoadFile" />
  </ElementType>
  </xsd:schema>
```

By shema files obtained the above, map the root element to the entity class, and other elements to attributes, and then form a local ontology. By the operations of integrating redundancy, and unifing, local ontologies generate global ontology. Figure 2, Figure 3, Figure 4, and Figure 5 are the corresponding local ontologies of Telemetry, RTU, PowerSystemResource and global ontology.

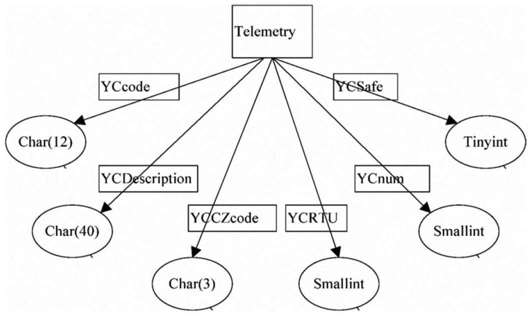

Figure 2. Telemetry Ontology

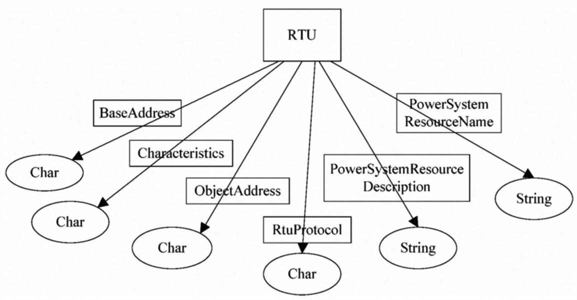

Figure 3. RTU Ontology

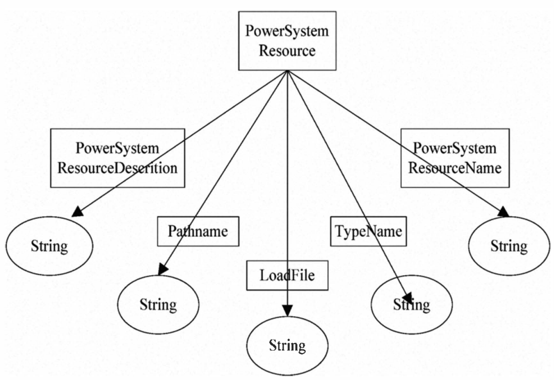

Figure 4. PowerSystemResource Ontology

The component of schema mapping: the schema of each local ontology must map, in order to exchange information. It connects components of ontology and query.

The component of query: it includes query receiver, Query Analyzer, and Query Builder. The query receiver accepts the user's query request from the application layer, then change the query to the Xquery-based standard form, finally, the query is turned over to the Query Analyzer for processing. Query Analyzer receives query from the upper, first maps thequery to the global ontology by mapping component; on the second, converts the global query into a corresponding local ontology form by analyzing relationship between the global ontology and local ontology, when the query reaches local ontology, it become query of the corresponding data source. Query Builder handles the returned results and changes them into XMl format, then presents them in a way that appropriates to the application layer through XSLT.

V. CONLUSION

In this paper, the use of xml and ontology technology has solvesd the problem of heterogeneous from syntax, semantics in power grid integration, at the same time, by combining technology of the components and middleware and cloud services we have achieved the Plug and Play function of power system components, and established the unified information platform of power system. This platform eliminates the "information island"of power system, so that power system has a higher degree in openess. System Features: higher openess, reusability, distribute supporting, easey of use. As the smart grid has just started, many applications are still in the theory research stage, not yet been realized. However, with the maturity of ontology and XML technologies, they will play an increasingly prominent role in information integration. Smart Grid will also be getting closer and closer to us.

ACKNOWLEDGMENT

Supported by “the Fundamental Research Funds for the Central Universities (09MR40)”.

REFERENCES

[1] S. Ram, J. Park. Semantic Conflict Resolution Ontology(SCROL): An ontology for detecting and resolving data and schema-level semantic conflicts, IEEE Transactions on Knowledge and Data engineering, Vol.16 no.2 February, 2004.

[2] D. Divan and H. Johal, “A Smarter Grid for Improving System Reliability and Asset Utilization,” Power Electronics and Motion Control Conference, August, 2006.

[3] W3C.XML Path Language(XPath),W3C Recommendation-[DB/OL]. (1999-11-16). http://www.w3.org/TR/xpath.html.

[4] Fhannelen, Jhendler, I Horrocks, etal. OWL Web Ontology Language Reference.World Wide Web Consortium. http:www.w3.org/tr/owl-ref. 2007.5

[5] Amazon.Amazon Elastic Compute Cloud Developer Guide[EB/OL].http://docs.amazonwebservices.com/AWSEC2/latest/ DeveloperGuide/, 2009-4-4.

[6] IEC61970, Energy Management System Application Program Interface (EMS-API) Part303: Common Information Model(CIM) SCADA[S]

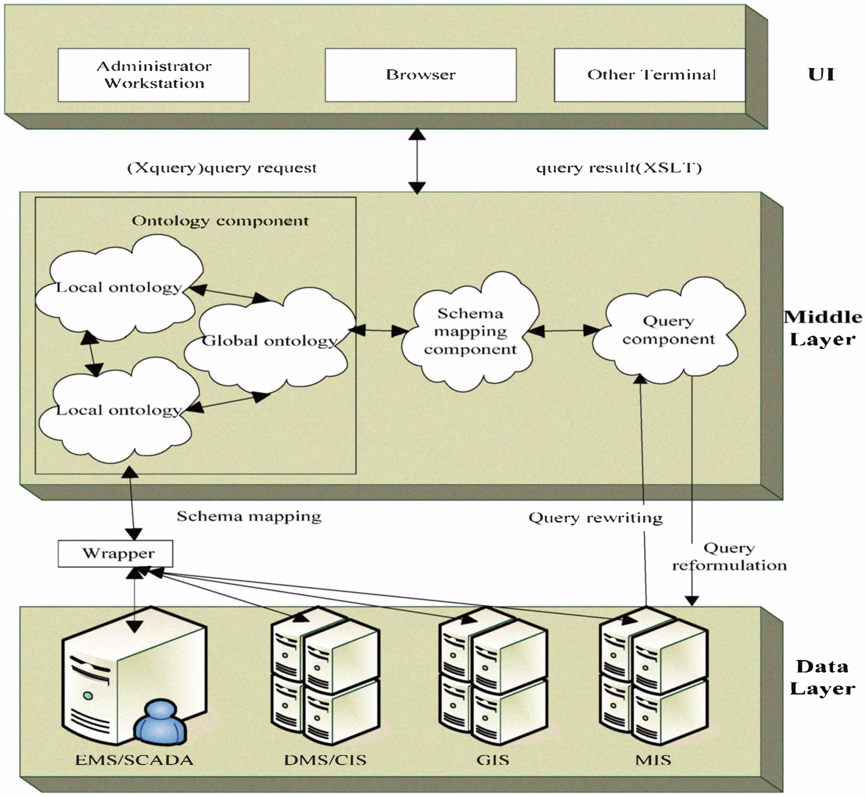

Figure 1. System model of Heterogeneous Data Integration

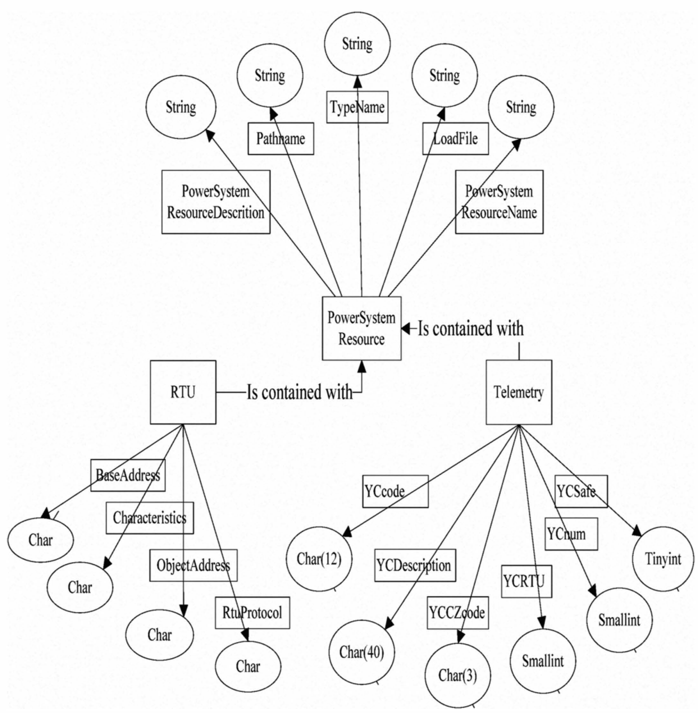

Figure 5. Global Ontology