INTEGRATING A GIS AND DMS SYSTEM USING OBJECT-ORIENTED TECHNOLOGY WITH AN EXTENDED EPRI MODEL FOR DISTRIBUTION

Erich Würgler
Telegyr Systems Corp.
Gartenstadtstr. 2a, CH-6301 Zug (Switzerland)
Tel: +41 41 724 41 67 – Fax: +41 41 724 51 78 – E-mail: erich.wuergler@ch.telegyr.com

Karl Herger
Telegyr Systems Corp.
Gartenstadtstr. 2a, CH-6301 Zug (Switzerland)
Tel: +41 41 724 34 56 – Fax: +41 41 724 51 78 – E-mail: karl.herger@ch.telegyr.com

SUMMARY

This paper describes the implementation approach of an integrated heterogeneous GIS and Distribution Management System with SCADA. It shows the integration of a long transaction, version managed data store in the GIS with a short transaction, mission critical real-time SCADA database. The role of standardization towards plug compatibility of applications is highlighted as well as Application Programming Interfaces within an object oriented object request broker technology environment.

GLOSSARY

AM/FM	Automated mapping /Facility management System
API	Application Programming Interface
CIM	Common Information Model
CORBA	Common Object Request Broker Architecture
DMS	Distribution Management System (includes SCADA)
EMS	Energy Management System
EPRI	Electric Power Research Institute
GIS	Geographical Information System, often with AM/FM functionality.
HIO	Human Interface for the Operation
IDL	Interface Definition Language
IEC	International Electrotechnical Commission
IT	Information Technology
OMG	Object Management Group
OODB	Object Oriented Database
PAS	Power Application Software
RTDB	Real-Time Data Base
SCADA	Supervisory Control And Data Acquisition
UCA	Utility Communication Architecture

INTRODUCTION

Deregulation leads to increased competition and therefore cost pressure. Moreover, it creates a business environment with ongoing restructuring, mergers, breakups and changing alliances. This forces utilities to be even more efficient in their business in order to survive.

Answers from utilities to this challenge may include:

- Extend SCADA to selected distribution substations and urban areas, to get better information at the lower voltage level (closer to the customer) and to allow for faster responses to outages.
- Use of advanced on-line applications for network management at the distribution level, (e.g., transformer load management, fault management, protection, equipment diagnostics, regulatory requirements). [1]
- Use of on-line and off-line applications for tasks like operational planning and optimization, maintenance and construction, network extension planning, meter reading and control, etc.

Many of those applications are already in use since years. They are optimized for their primary task. However, they often

- Represent islands from a corporate IT viewpoint.
- Are based on different data models, which represents a major obstacle for adding "off-the-shelf" applications and for sharing data.
- Contribute significant cost due to their specific way of operation, maintenance and especially data maintenance.

The need for a homogeneous way of operation (same look & feel), improved workflow among the applications, homogeneous kind of maintenance and redundant-free data entry is obvious.

Nevertheless, it is not practical to replace the legacy applications at once. Nor should the utility go for a monolithic solution, as this would not fulfill the flexibility required in today's fast changing environment. An open and modular approach is required instead, which allows in a cost-effective way to integrate existing and future applications and to react to the changing environment.

BENEFITS OF AN INTEGRATED GIS AND DMS

The classical purpose of a GIS is the graphical “as built” documentation of a network:

- Maps
- Facility management
- Planning

The classical purpose of a DMS is the operation of a network, based on its current state:

- Controlling/Monitoring
- Trouble management
- Optimization
- Security

Both systems work on the same network, with a large overlap in the data area. Integration of GIS and DMS systems therefore enables the following potential benefits:

- Cost reduction in data maintenance.
- Workflow optimization from planning to installation, operation and maintenance.
- Cost reduction for system procurement and maintenance (total cost of ownership)
- Improved application interoperability between planning and operation.
- Process optimization (Remote and closed loop control)

ARCHITECTURE OVERVIEW

Implementation Goal

The implementation of the system is driven by the goals:

- Combine the strengths of a GIS system and a distribution management system, in order to reduce data maintenance costs, while maintaining the real-time features for network operation, and improving the operation efficiency.
- Provide an open platform for cost-effective integration of multi-vendor applications, and for easy integration into existing IT infrastructures and business processes.

System Architecture

The system uses both GIS technology and real-time distribution management system technology (See Figure 1).

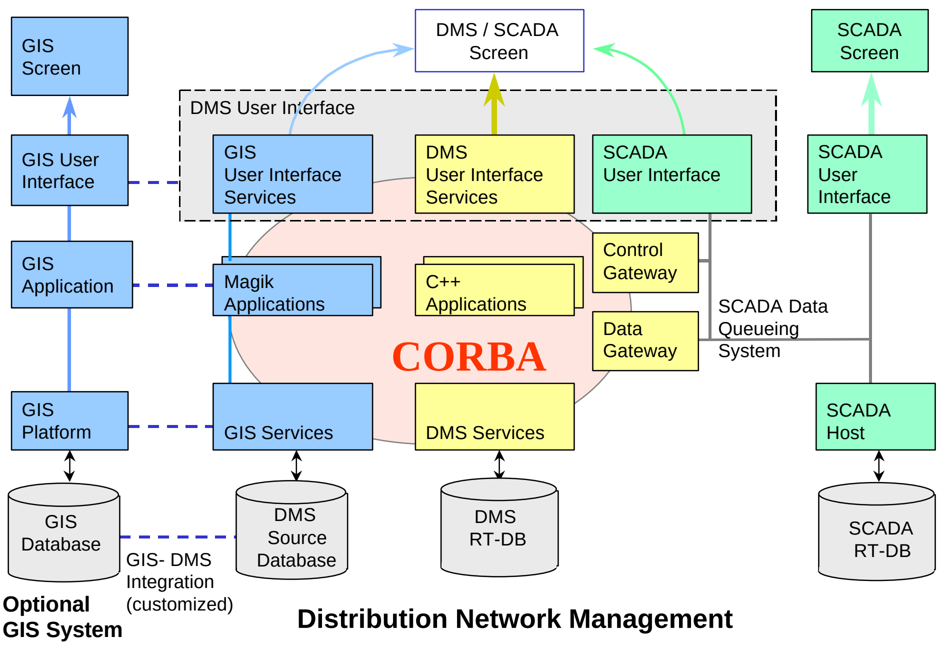

Figure 1: Architecture Overview

A version managed datastore is used as source for all the tabular and schematic data. Data is entered either through an interactive graphical editor, or it can be imported from existing data sources and later kept in synch. Alternatives in the database allow to maintain the current network “as built”, as well as future versions for planning, or past situations for later analysis.

An object-oriented database (OODB) provides a common view to the current network state for all network operators, with spontaneous updates of changes to all connected operator workstations. This common view is called the real-time context. Study contexts, which are local to an operator, give the possibility to work with network loads, operating conditions or network configurations that are different from the current situation, without interfering with the real-time context.

The user interface is implemented in Magik[1] for the operator interaction, providing fast prototyping and easy customization to specific user needs. A drawing engine, optimized for fast display refresh and written in C++, presents the network schematic diagrams. It retrieves the real-time state of the objects from the OODB for graphical state representation.

Gateways to a SCADA system allow the reception and processing of real-time status and event information, as well as supervisory control to remote network objects.

A CORBA based Application Programming Interface (API) supports the communication between the system components, and it allows the integration of other applications into the platform.

[1] Magik is an application language close to Smalltalk

DATABASES AND DATA MODELS

A typical customer will have most of the required data already available in electronic form. One must assume that each customer uses its own data model, which can be quite different from one customer to the next one, and between departments within a utility. One possibility would be to always adapt our system's data model to the customer's model. This would require changes in the applications in most cases and could result in high software engineering costs. Another approach is to use a standardized data model within the distribution management system, and to convert existing data into this model. This yields a stable base for the applications, and it makes it easier for system upgrades. The architecture supports to store (or reference) data in the source database both in the internal and in the customer specific (facility) data model. The import (and later synchronization) between the internal and the external model can be accomplished using the GIS tools. Additional, application domain specific tools provide enhanced data validation.
The question is, of course, which data model to use internally.

EPRI Common Information Model

The Electric Power Research Institute (EPRI), Palo Alto, California, runs a project called Utility Communication Architecture (UCA). One result of this project is the Common Information Model (EPRI CIM). The CIM is an object oriented data model developed originally for Energy Management System applications. It has been submitted to an International Electricity Commission (IEC) working group as proposal for international standardization. The section of the EPRI CIM that models the basic electrical network objects, like, e.g., breaker, fuse, transformer and their relationships, has been taken as base for the data model. It has been enhanced for specific distribution needs, like, e.g., feeders, junctions, distribution loads, and will be submitted to EPRI and IEC as proposal to enhance the standard.

Version-Managed DMS Source Database

The relational database manages all the "static" data in the system. It represents the system "as built". The network connectivity is derived during data entry from the graphics using GIS techniques. Hierarchical alternatives are used to manage long transactions (e.g., capture of a network extension, and its simulation and test until it is put into operation). The "top" alternative always represents the current version of the "as built" network, while sub-alternatives store future extensions or past configurations, for planning, analysis, or training. Different database partitions separate source data in the facility model from source data in the EPRI model.

Object-oriented Real-Time Database

This database contains the current network state. Using transactions and locking, it synchronizes the database access from multiple workstations with the server database and guarantees consistency. Change notifications are distributed to all connected workstations. Caching techniques improve the performance in a distributed environment. The real-time database is built by extracting data from the relational source database. The common real-time context is kept in synch with the "top" alternative of the source database. Study contexts (see page 2) are related to the top or any other source database alternative.

The DMS Services that connect the applications to the Real-time database are an essential component in the whole system. They acts both as client to the real-time object-oriented database and as server to all its API clients like, e.g., the drawing engine, the user interface, the SCADA gateways, or network calculation applications. Their main functions include

- Distributed, cached read- and write access to the real-time database
- Transaction handling, locking
- Database event handling and notification to subscribed API clients
- Support of real-time and study contexts, with synchronization to source alternatives
- Basic DMS operation engine (on-line network topology calculation)

Together, those elements support key database requirements in distribution management:

- Long transactions for network extensions and planning in data maintenance.
- Short transactions for real-time state updates in network operation.
- Contexts for studies in planning and operation.

APPLICATION PROGRAMMING INTERFACE (API)

The EPRI reference architecture [2] defines an application interface and a common data access interface between the applications and the platform. In our solution, this functionality is implemented through CORBA. Working groups from EPRI and IEC (where we are active members) and the Object Management Group (OMG) are working towards standardization of those buses, which are intended to serve as a general utility integration bus.

API functions

The API enables an application to talk to the source database, to the real-time database, to the user interface, and to other applications. The main functions include:

- Get technical data (e.g., line admittance from the source database)

- Get status information (e.g., supply state of a distribution load)
- Update real-time data (e.g., set breaker to closed)
- Topologic queries (e.g., get all device IDs downstream of a fuse)
- Subscribe to event (e.g., a state change in the real-time database)
- Receive an event notification (e.g., when SCADA has updated a breaker state)

The API is defined in OMG's Interface Definition Language (IDL) and implemented using a commercially available Object Broker.

Object granularity

A key element when designing an object-oriented API is the decision about which objects are to be exposed on the interface as CORBA objects. An intuitive approach would be to expose every network object, e.g., every breaker, line, source as a CORBA object, and to define read, write and control methods on those objects. This approach is sometimes referred to as "fine granular" or "little" objects. Its main drawbacks are performance and the difficulty to embed the access methods in transactions. Measurements on Unix have shown that one cannot expect to handle sufficient CORBA requests per second, and that (unless limited by network bandwidth), the rate drops not significantly with increasing packet size. One can expect a similar behavior on the Windows/NT platform. This leads to an alternate approach, using so called "coarse granular" or "big" objects. The object exposed to the API is a server object, or a component, that accepts requests on behalf of the "little" objects, executes the requests, and returns the results to the client. Across CORBA, the data has to be transmitted in the right format and packaging, to reduce the CORBA traffic, and the server component can embed the operations in a transaction.

API and CIM benefits

The API is used by the basic platform services (user interface, rendering), by the SCADA gateways, and it is the common interface for all DMS applications.

Using an open technology like CORBA to implement the API gives freedom in choosing an application language, and it fosters the definition of clean interfaces (IDL). It allows for easy system scaling, as the different applications address each other through a broker, independent of their physical location.

Together with the EPRI Network Control Center reference architecture, the CIM should finally enable "plug compatible" applications for distribution management systems. Our solution represents a good step towards this target. The API and the CIM based data model can significantly reduce the costs and risks involved with integrating applications onto such a platform.

Utilities will profit from a broader selection of available applications and from reduced implementation costs.

SYSTEM INTERFACES

User Interface

The user interface is enhanced with a fast rendering engine for real-time status display. It supports the data maintenance, the network operation (for remote-controlled and non remote-controlled objects), and GIS and SCADA interaction. The ongoing development in the IT area will lead to further progress in user interface harmonization, making user's life easier through common look-and-feel for diverse applications.

SCADA gateways

Two gateways were implemented for coupling with a SCADA system. One gateway, the Data Gateway, feeds data from the SCADA system into the DMS real-time database, by subscribing to SCADA objects. A second gateway, the Control Gateway, supports control request from DMS to SCADA, for, e.g., device operations or user interface interactions. A correspondence list between SCADA and DMS object identifiers is maintained in the DMS source database. The gateways represent a well-defined interface between the DMS and the SCADA functions. Each gateway consists of three parts: A first part uses the DMS API to communicate with the DMS system. A second part uses the SCADA specific API to communicate to the SCADA system, and a third part connects them and translates the requests from one format into the other one, driven by transformation rules. This architecture allows an efficient adaptation to different legacy SCADA systems. Our approach with a standard data model, open interfaces and a service layer enables us to both integrate legacy systems as well as to achieve a homogeneous solution with our own integrated SCADA functionality.

GIS coupling

One purpose of coupling the DMS system with a GIS system is to reduce data maintenance costs (single source of data entry). For reasons of responsibility, flexibility, performance, security, availability, such a coupling will be quite loose in most cases. We can expect many flavors, depending on the customer's GIS supplier, and whether the GIS system is installed prior, in parallel, or after the DMS system. Customer workflows and procedures, organizational responsibilities, preferences, data model and data volume can influence the coupling in many ways. Consequently, a coupling to an existing GIS system usually needs to be customized for the particular case.

In general, the DMS system has to be ready to accept data updates in smaller or larger batches, to determine the differences, conflicts, and missing data. In some cases, the DMS system might also have to send updates from its database to the GIS.

The applied technology provides excellent support for such functions, with its version management, the difference streams, and the query possibilities.
Data can be validated and tested in the DMS operation environment, using a study context. Testing in this real environment improves the stability and security of the on-line system operation.

INTEGRATION ISSUES

DMS Model versus GIS Model

Data modeling for GIS is different from the typical modeling for distribution network operations due to the need to model spatial and topological relationships [3]. On the other hand, DMS models have some additional complexity when compared to GIS models, due to the need to model real-time aspect, such as current connectivity state of the distribution network. The basic purpose of a GIS is the (geo)graphical "as built" documentation of a network. The purpose of a DMS is the operation of a network, i.e. presentation of the current state of the network "as is", and the control, and it enables a higher level of automatization. DMS and GIS models typically share most of the static and structural information. For static information, the DMS model contains a set of data that should be derived from a GIS database. This task, in some instances, might be fairly complex due to the fact that granularity of information contained in a GIS database is typically higher (or at least different) than the corresponding information in a DMS Database. For example, poles and line segments between poles are represented in greater detail in GIS than it is required in DMS. The GIS representation of a line might contain information for 5 poles, 4 line segments and 1 load, while the corresponding DMS description has 1 line segment in a given wire arrangement and 1 load. Such data reduction is a necessity to give the operator a clear network overview and to reach the required on-line performance for network calculation.

To automatically derive the DMS data from GIS, a rule-based translator that uses certain network reduction rules may be used.

Single source of data entry

To avoid duplicate data entry in order to reduce costs and to improve data consistency, the idea of a centralized source database for all utility applications looks tempting. Compared to the situation where each application requires its own data definition, the savings that could be achieved are immense. However, the different departments in a utility with their applications have quite different views of the network and different needs, as we have seen in the previous section. To define a database that fulfills the data requirements for all applications of a utility looks like a never-ending task.

A more realistic approach for a utility is presented in [4] and also supported by [1]:

- Identification of the common elements between all the systems to be considered (though their representation might differ).
- Inventory of the data to be shared.
- Assignment of data responsibility ("data owner") to each system
- Definition of interfaces
- Definition of data exchange rules (permission, performance, synchronization, validation, version management)

The resulting objects from the analysis in [4] are a very close match to the "wires model" objects in the EPRI CIM [5].

Object oriented techniques make such an approach feasible. Interfaces between the different systems can be defined independent of the physical data storage (in contrast to a SQL based solution, where the SQL queries heavily depend on the physical data layout). It is also possible to encapsulate legacy systems with "wrappers" to make their data accessible through such interfaces.

CONCLUSION

The described implementation is a platform with a powerful and future oriented architecture. The real-time database combined with a high performance topology processor allows the control room operator to have a fast and clear view of the current state of the network, to highlight deviations to the normal state, to easily locate network faults and subsequently restore the network. The system supports the ongoing introduction of remote control and automation in medium voltage networks. Application Programming Interfaces provide means to implement in house and 3rd party applications. The reliable technology of APIs has been in use in our Network Management Systems since the early 1980's and was a valuable contribution towards open systems especially when migrating to new software versions. The high reliable and high performance distribution management technology combined with the version managed database technology from GIS efficiently supports the utility in the whole workflow optimization, and in the network operation, extension and maintenance. The design would also allow to include links to commercial applications related to DMS operation. Common data capture and maintenance for DMS and GIS is a substantial cost benefit beside reduced training and engineering costs and savings of commonly used workstations and peripheral devices.

REFERENCES

[1] Draft IEC 61968-1 1998: System Interfaces for Distribution Management – Part 1: Interface Architecture and General Requirements

[2] Preliminary Draft IEC 61970: Energy Management System Application Program Interface (EMS-API) – Part 1: CCAPI Guidelines

[3] Stipe Fustar, "Towards a Common Information Model for Distribution Operations Management", *DA/DSM February 1999*

[4] Jacques Plumer, Georges Backes, Gilbert Suter, "A Common Data Model at CEGEDEL, The first step towards integrated network operation and management", *DA/DSM London, October 1998*

[5] Draft IEC 61970: Energy Management System Application Program Interface (EMS-API) – Part 301: Common Information Model (CIM)