Flexible Data Model Architectures for Real-Time Power System Control

Santiago Grijalva
santiago@powerworld.com
PowerWorld Corporation, IL, USA

Abstract

This paper illustrates how new object-oriented programming techniques and flexible data structures can be utilized to make off-line power system analysis applications available to the EMS environment. De-mystifying EMS systems, applications, and data structures we describe the basics of integrated architectures that have been implemented at some of the largest control centers in the United States. These new systems are allowing operators and EMS engineers to use powerful analysis and visualization tools to support the needs of modern power system operation.

1. Introduction

Modern power system control center operations face numerous challenges originated in the economic forces of deregulation and technology development. These forces have resulted in more stressed operation close to security limits in power systems around the World. Past blackout events in New York and Italy reflect the complexity of real-time control and the need to enhance existing systems with tools that can predict dangerous conditions and alert the operator to take corrective actions [1,2]. It also points out the need of better integration between real-time operations and system planning.

Historically, real-time operations and system planning have been separated both from the enterprise process point of view and from the technologies therein utilized. There were few processes “in the middle”. This separation created numerous technological difficulties to power system and computer engineers. The real-time features needed by the data acquisition and processing functions of EMS systems forced EMS vendors to use real-time operating systems. The planning environment on the other hand adopted a much less expensive personal computer approach. The proliferation of generic personal computers resulted in fast evolution of new algorithms, visualization and graphical user interfaces. Ironically, the more expensive real-time environment did not evolve as fast as the personal computer industry. As a consequence, some analysis and visualization features that are available in the planning environment have not been implemented in real-time systems. In addition, the core algorithms in most operative EMS systems have not migrated to object oriented programming and maintain their Fortran and C implementations, considerable limiting technological development.

2. EMS and Planning Models

At the core of this process separation are the power system EMS and planning models that were adopted by the industry. While real-time EMS applications model the complete topology of the power system at the breaker-node level, the planning model is a less detailed bus-branch model that was more portable and easy to use in personal computers. Core applications, such as the power flow and contingency analysis, which in essence are meant to model the same system and determine the same quantities require different implementations in the planning and in the real-time environment. In addition, since most of the technology utilized in real-time systems was proprietary, for the longest time there was no mechanism to compare models that were supposed to mimic the same physical system and to determine their consistency. Even with great advances in networking, open operating systems, databases, and the internet, there are few control centers that work with fully compatible planning and real-time cases. This was pointed out as a strong limitation for the post-mortem analysis developed by task force that investigated the New York 2003 blackout [1].

In the EMS model, system elements and devices are specified by names. Some EMS systems use a three-level naming convention STATION–VOLTAGE LEVEL–DEVICE ID to represent system elements and devices. For instance a name ADAMS 13.8KV GEN1 denotes a unique generator in the system, which is connected to a 13.8kV node in the Adams station. For branches, a convention is FROM_STATION–VOLTAGE–TO_STATION. For instance, line ADAMS 230KV ALDENE. This type of naming convention allows operators to easily recognize different elements for alarm and action processing [3]. On the other hand, the planning (bus-branch) model consists of buses and branches which are identified by bus numbers. All devices are attached to buses and adopt the corresponding bus number.

Deregulation and electricity markets have created a tremendous need to overcome this planning and EMS separation and have created great integration possibilities. There are many examples of processes where EMS and planning compatibility would be of great service [4]. Seamless exchange of models between these two environments would result in significant benefits, including enhancement of operating practices, increase in the capability to predict insecure conditions in the system, better tools to identify economic opportunities in the electricity markets as well as to monitor market efficiency, unification of control center technologies, and savings in time and cost of personnel training.

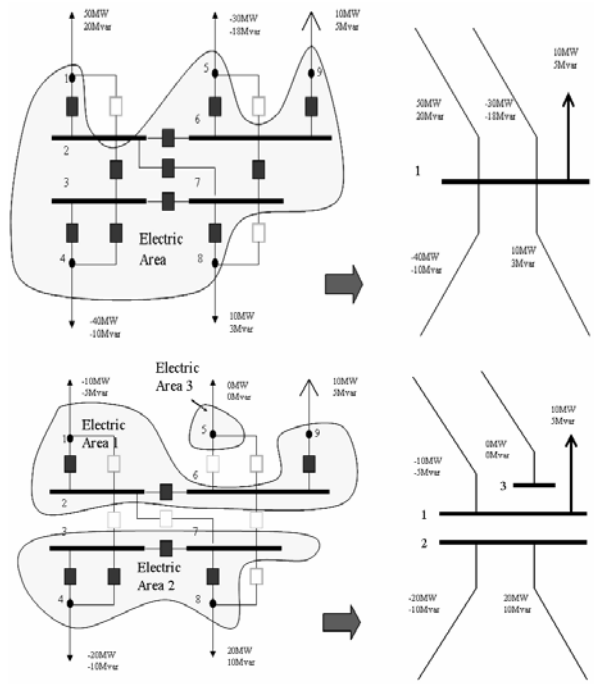

Figure 1: Groups of EMS nodes connected through closed breakers form buses in the planning model.

3. Topology Processing

In this section we describe Topology Processing, the algorithm utilized to dynamically generate a smaller planning representation of the EMS system and how references that allow mapping quantities from the EMS model to the planning model are kept.

Figure 1 shows a typical configuration of a small station. Transmission lines are connected to the bus bar sections through circuit breakers. Given a breaker status configuration we can determine groups of nodes that correspond to the same electric point and that are called *Electric Areas.* Each electric area in the EMS model corresponds to a bus in the planning model. While it is possible to generate a planning model from the EMS model and the breaker statuses, it is not possible to know the EMS model from a planning case. When a planning case is created from the EMS model, it is customary to number the resulting buses sequentially. Because the number of electric areas changes dynamically as breakers change their statuses over time, the number of a given bus does not remain the same from one data retrieval to the next. While off-line planning cases keep the same bus numbers, any planning model generated from the EMS will have different bus numbers. This is why it is difficult to compare EMS and planning cases.

There are two types of topology processing [5,6]:

a) *Full topology processing*, in which the planning model is created from scratch
b) *Incremental Topology Processing*, in which the breaker statuses that have changed since the last data update are identified and the electric areas and buses are split or merged correspondingly updating the topology of the existing planning case.

In full topology processing, the planning model is re-created during each EMS sequence. The identification of the electric areas of the system is implemented with a recursive function whose implementation is summarized as follows:

```
routine FullTopologyProcessing
set all nodes as not visited
for each node do
   if node was not visited then
      create a new EA with this node inside
      add EA to the list of electric areas
      setElectricArea(node,EA)
   end;
end;
```

```
function SetElectricArea(node,EA)
set node as visited
while there is a connection from the node do
  if the connection is a closed breaker then
```

```
    OtherNode = connection.OtherNode
    if OtherNode was not visited then
      add OtherNode to node list in EA
      add connection to breaker list in EA
      SetElectricArea(OtherNode,EA)
    end
  end
end
```

This recursive function is fast and can identify all the areas in very large models in less than a second. At the end of the process we obtain a list of electric areas, each one of them containing a list of nodes and a list of breakers.

The second step in full topology processing is to merge the nodes that are inside each electric area. The following function summarizes how this is done:

```
routine NodeMerging
for each ElectricArea do
  move all gens, loads and shunts to pnode
  move all the node pointers to pnode
  remove the breakers inside the ElectricArea
  update connect branch terminal to pnode
  update pointers of remotely regulated buses
end
```

At the end of this process, all except one node from each electric area has been removed. This preserved node (pnode) has now attached to it all the devices that were previously attached to other nodes in that electric area. All closed circuit breakers have been removed. All external transmission lines, series devices, phase shifters, etc that had a terminal in a node of the electric area have moved its terminal to the pnode. All other pointers of remotely regulated buses, etc are updated to the pnode.

The nodes that have been preserved in the model become buses in the planning model, and thus there are no circuit breakers, isolators, junctions, or bus sections in the model anymore. Typically the planning model has about 10% of the number of nodes present in the EMS model [7,8].

4. Incremental Topology Processing

This algorithm updates the planning model based on the updated circuit breaker status during data retrieval. Suppose hypothetically that all circuit breakers of the power system were closed. Then the electric areas would be as large as possible, but they will still be limited by the boundaries of the voltage level within a substation. Either a transformer or a transmission line will separate a group of breakers from the other. Let us call these special electric areas a *breaker group*. By running full topology processing assuming that all circuit breakers are closed we can determine all the breaker groups in the system. We note that these breakers groups are constant unless the EMS model is modified with new devices or changes in the physical topology of the system. Hence, the breaker groups can be determined during database generation.

In order to update the planning model due to changes in breaker statuses only those breaker groups that include breakers that changed their statuses need to be re-processed. Incremental topology processing can then be implemented by running full topology processing only on those breaker groups.

```
routine IncrementalTopologyProcessing
for each breaker do
  if breaker status changed then
    set breaker group as modified
  end
end
for each node do
  if node is in a modified breaker group then
    set node as not visited
    restore node breaker group
  end
end
run FullTopologyProcessing
(without setting nodes as not visited)
```

5. Re-Usable Data Structures

The EMS model can be stored in the same data structures that are used to store the planning case. In order to realize this, we should note that that nodes, junctions and bus bars can be stored as buses, while circuit and section breakers can be stored as (zero impedance) branches. Having the EMS and the planning model in the same data structures has a number of advantages:

- It facilitates converting the EMS model into a planning model.
- It allows comparing existing planning models with models derived form the EMS.
- It allows sharing the models for use by other applications.
- It simplifies data engineering.
- It facilitates software maintenance and software enhancement.

EMS device IDs are preserved in the case as custom string fields. This allows keeping track of the original device names, mapping results after the solution, and visualizing devices in both systems.

5.1 Source and Copy Objects

A basic technique within topology processing is to use copy and source object at the model and object level. This is straightforward if an object-oriented architecture is used for the implementation. The references to source and copy objects are initialized during object creation and removed before object destruction. The power system model will have a pointer to a copy and a source model.

Prior to full topology processing a copy of the EMS model is created and kept in memory. Topology processing then acts on the copy model which gets "consolidated" into a planning case. Objects in the copy model have only source objects, while objects in the source model have only copy objects. References from objects in the source case to objects in the copy case are maintained during topology processing. Handling source and copy objects allows for very fast mappings. For instance, if one needs to determine what is the bus into which an EMS node was consolidated during topology processing, one has to look at the EMS node's copy node. On the other hand, if one solves the power flow on the planning case and wants to map a generator's MW output to the EMS model, on just passes the field values to the source object. This is illustrated in Figure 2 for node1, which has been "consolidated" into node 4 and for generator B.

Using source and copy objects has the advantage that one can create copies from the copy, etc. Let us consider the case of the user needing a case for study model. After topology processing has taken place, both the EMS and the planning models are available in memory. If changes in study mode do not include modifications of the system node/breaker topology, then a copy of the planning model can be made. Otherwise a copy of the EMS full model can be obtained in order to initialize the study mode session. This architecture also allows the creation of multiples copies for different applications, parallel processing, etc.

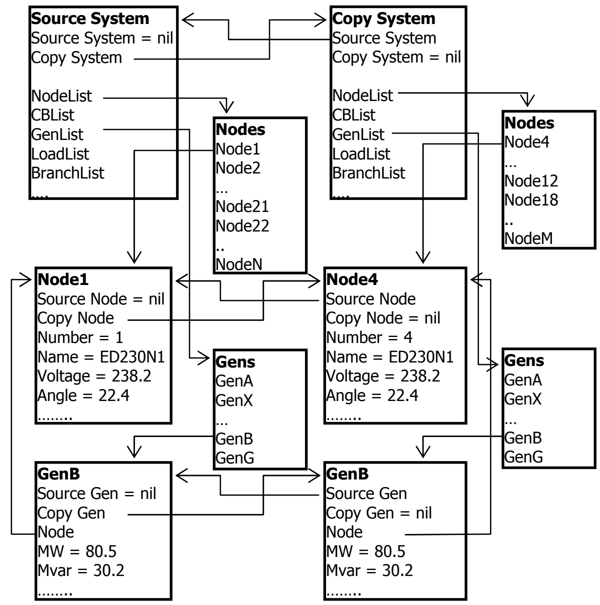

Figure 2: Mapping from Copy and Source Objects

5.2 Restoring Functions

During incremental topology processing, only those breaker groups that have changed are reprocessed, i.e., the closed breakers are determined and the corresponding buses are obtained by merging the resulting electric areas. Before this occurs, the electric group needs to be restored, i.e., all the original nodes of the electric group are re-created together will all the circuit breakers inside the group, and all device references are changed from the pnodes to the original nodes. This restoration process uses copy and source objects and takes place as follows

1. All breakers in the EMS model that have changed status since the last scan are flagged:
   Breaker.StatusChanged = True.
2. Flag breaker groups that need to be reprocessed. Since EMS breakers know to which breaker group they belong we can go through all the breakers:

   for all breakers **do begin**

       if Breaker.StatusChanged = True **then**

           Breaker.BG.MustProcess = True

   end

For each breaker group that must be processed, do the following

3. Copy to the planning model the nodes of the breaker group, except those that are electric area preserved nodes (pnodes). The nodes that were copied do not have any devices connected to them.

4. Copy all the breaker group breakers to the planning model, except those that may be already in the planning case. A breaker may still be in the planning case since it was opened during the previous scan and may be separating two electric areas in the electric group.
5. Move the breaker group devices, which are connected to the electric areas pnodes in the planning model to the recently restored nodes. Generators, loads and shunts are connected to a single node. Thus the original connection node for these devices is determined using the source object:
   Device.Node = Device.SourceDevice.Node.CopyNode
   For transmission lines and transformers it needs to be determined which end of the branch must be moved to the original node. This is easily done by finding whether the terminal node is actually in the corresponding breaker group that is being restored.
6. After all the breaker groups have been restored in this manner, topology processing takes place on the restored breaker group. This is a very fast approach that manages to update the planning model topology.

5.3 Retrieval Architectures

A basic design goal of power system applications is to make them compatible with different EMS database platforms ranging from flat text files and relational databases to historical databases [9,10]. The application must be designed to retrieve a *snapshot* of the system, i.e., a complete description of the power system at a certain point in time, usually equivalent to the state estimator solution or a complete set of the process measurements, indications, and events. This goal is achieved with generic objects capable of interacting concurrently with different data source types. To the operator it is irrelevant whether the data comes from one or more data sources of different types. The data sets are unified through the model and diagrams managed by the application, facilitating discovery of relevant information for analysis.

A related goal is displaying information of particular types without the need to model the entire real-time snapshot. For visualization purposes, the data sets of the system snapshot are typically grouped by:

- Voltage level, for instance, only 230kV and above.
- Geographic region
- Object type: node, substations, zone, or control area.
- Level of detail, for instance substation object or substation topology diagrams at the node/breaker level.
- Thematic diagrams such as voltage contouring, marginal price profiles, reactive power reserves, etc.

Often, control center operators require a specific model that requires only a small subset of the entire real-time snapshot. In order to achieve this, the visualization application uses the concept of aliases. An *alias* is data point name that relates the value in the data source to a graphical object in the diagram. As an example, suppose that a certain 138kV bus named ADAMS 138KV NODE 1 in the system has a real-time voltage of 136.3KV. We can define an alias called ADAMS 138KV NODE 1.VOLTKV where the first part of the name is typically the EMS name and the suffix is the quantity type. The datasource contains a table of this alias together with the real-time value. It also contains a table of alias subscriptions. An *alias subscription* is a mapping from a graphical object field to the alias. For example, the following subscription:

3465, BusKVVolt, ADAMS 138KV NODE 1.VOLTKV

states that the KV voltage field of bus 3465 drawn in the oneline diagram will display the kV value associated to the alias ADAMS 138KV NODE 1 which corresponds to the actual bus in the system. Figure 3 illustrates the main concept of the alias subscription architecture

The use of aliases and alias subscriptions completely decouples the EMS data from the objects. In this manner, models with as little as few key devices can be built regardless of the EMS data model. A second advantage is that by creating fictitious devices, the objects in the diagram do not have to correspond to actual devices. Suppose that only the flows through a set of phase shifters need to be displayed in a diagram. Then phase shifters graphical objects can be drawn on the diagram with arbitrary from and to node numbers and names. The fields of the phase shifter device will subscribe to the corresponding phase shifter aliases to display the correct values of for instance MW flow and phase angle. If the voltages at the from and to ends of the phase shifters are to be displayed, the association between these bus objects and the actual devices must be established.

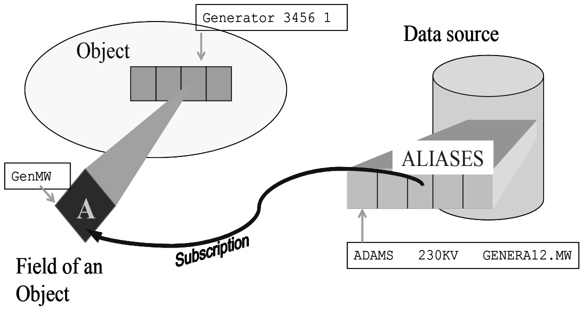

Figure 3: Alias Subscription Architecture

Other advantage of aliases is the possibility to subscribe to alias expressions. This allows modeling all sort of calculated values. One of the key challenges when using alias and alias subscriptions is the resolution of the alias name when reading real-time data. Special searching algorithms are utilized to make the search fast and comparable to a sorted list search. Common speeds of data reading are about ten thousand data points (measurements or indications) in 0.1 seconds.

6. Application to Real-Time Functions

Figure 4 shows a typical implementation of advanced applications within an EMS environment. The larger the clock-wise feedback loop the more complex and advanced applications are involved to provide a deeper understanding of power system conditions. The core of the application functions is the *State Estimator*, which provides the most probable state of the actual controlled power system based on a set of redundant process measurements and indications. As a result of the estimation process a solved power flow case can be obtained. Both the estate estimator and the real-time power flow act on a planning model.

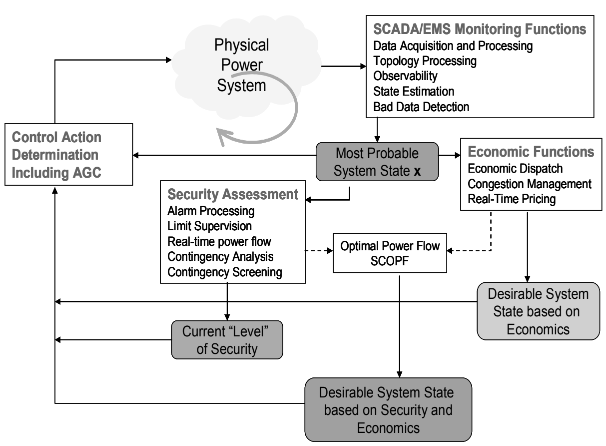

Figure 4: EMS Advanced Applications Implementation

The state estimation is the principal component of what is called the EMS Sequence [11-13], which includes:

- Gross Error Detection (GED)
- Topology Processing
- Observability Analysis
- State Estimation
- Bad Data Detection

A dialog showing the summarized results for the state estimation sequence is shown in Figure 5.

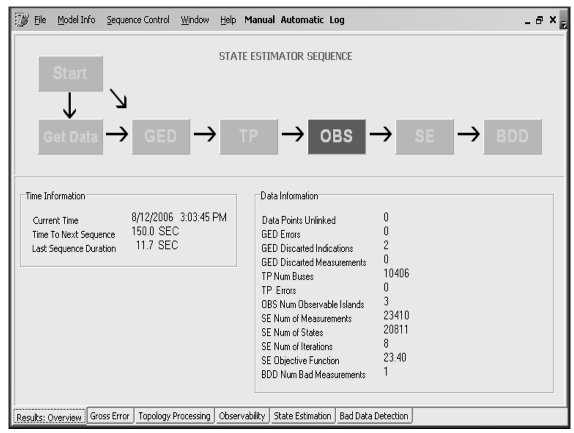

Figure 5: EMS Sequence Result Dialog

Once a state estimator solution is obtained, one can take advantage of flexible data structures described in the previous sections in order to use applications that were previously limited to the off-line use into the online environment. These applications typically include:

- Operator Power Flow
- Sensitivity Analysis
- Contingency Analysis
- Economic Dispatch
- Optimal Power Flow and SCOPF
- Marginal Pricing
- Voltage Stability Assessment
- Transfer Capability Analysis
- Market Applications

All these applications would work over the planning representation of the EMS model, which is continuously being updated by the incremental topology processing function. Solutions of the different applications can be readily applied to the EMS model for visualization purposes using the existing references to the source objects.
This integration of off-line with real-time systems has numerous advantages to the utility business processes and to the overall operation of power

systems. Large control centers in the US are starting to take advantage of these flexible data structures and software to combine the power of off-line and online applications to enhance their performance.

7. Conclusions

EMS models can be stored and analyzed using the same data structures and routines used in planning models. One can develop planning models using topology processing algorithms.

Flexible data structures including source and copy objects, restore functions and retrieval methods can be utilized to achieve flexibility to move results and data back and forth between EMS and planning models.

Using these flexible data models analytical applications that were previously limited to the off-line system can be utilized within the real-time environment to boost system analysis and enhance power system operations. These new architectures are being used successfully in large control centers in the US.

8. References

[1] US-Canada Power System Outage Task Force, "Final Report on the August 14, 2003 Blackout in the United States and Canada: Causes and Recommendations", US Department of Energy, April 2004.

[2] Autorità per l'Energia Elettrica e il Gas, "On the Events of September 28th, 2003, Culminating in the Separation of the Italian Power System from the other UCTE Networks", April 22nd, 2004

[3] E. Vaahedi, M. Shahidehpour, "Decision Support Tools in Restructured Electricity Systems: An Overview", IEEE Transactions on Power Systems, Vol. 19. No. 4, November 2004.

[4] T.J. Overbye, J.D. Weber, and M.J. Laufenberg, "Visualization of Flows and Transfer Capability in Electric Networks", Proc. 13th Power System Computation Conference, Trongheim, Norway, June 1999. pp 420-426.

[5] B. Nilsson, "Notes on Applications Functions", ABB Network Partner, Vasteras, Sweden, 1997.

[6] ABB "Network Applications Real-Time Control Curse: T50", ABB Network Partner, Vasteras, Sweden, 1998

[7] A. Bose, K.A. Clements, "Real-Time Modeling of Power Networks", IEEE Proc. Special Issue in Computers in Power System Operations, Vol. 75, No. 12, pp. 1607-1622, Dec., 1987.

[8] A. Monticelli, A Garcia, "Modeling Zero-Impedance Branches in Power System State Estimation", IEEE Transactions on Power Systems, Vol. 6, No. 4, pp. 1061-1069, July 1993.

[9] T.J. Overbye, D.A. Wiegmann, and R.J. Thomas, "Visualization of Power Systems", PSERC Final Report [Online] Available at www.pserc.wisc.edu, 2002.

[10] Y. Sun, T.J. Overbye, "Visualization for Power System Contingency Analysis Data" IEEE Transactions in Power Systems, Vol. 19. No. 4, November 2004, pp. 1859-1866.

[11] M.P. Selvan, K.S. Swarup, "Object Methodology: Method and Design for Topological Processing", IEEE Power and Energy Magazine, Vol. 3, No. 2, Jan-Feb, 2005, pp. 18- 29

[12] A. Monticelli, "Modeling Circuit Breakers in Weighted-Least Squares State Estimation", IEEE Transactions on Power Systems, Vol. 8, No. 3, pp 1143-1149, Aug. 1993.

[13] A. Monticelli, "State Estimation in Electric Power Systems: A Generalized Approach", Kluwer, Boston, Massachusetts, 1999

Biography

Santiago Grijalva obtained the Electrical Engineer degree from EPN-Ecuador in 1994, the M.S. Certificate in Information Systems from ESPE-Ecuador in 1997, and the M.S. and Ph.D. degrees in Electrical Engineering from the University of Illinois at Urbana-Champaign in 1999 and 2002, respectively. He completed the Post-Doctoral program in Power and Energy Systems at the University of Illinois in 2004. From 1995 to 1997, he was with the Ecuadorian National Center for Energy Control (CENACE) as EMS engineer and Head of the Software Department. Since 2001 he develops advanced power system applications for PowerWorld Corporation. He has been a consultant for major government agencies and utilities across the US. During 2005 and 2006 he was an invited professor at EPN where he taught graduate courses on advanced power system analysis and power system economics. His areas of interest are real-time systems, power system computational algorithms, and electricity markets.