

Network Modeling and Model Updates for Large Interconnected Transmission Networks

*Jay Britton, Fellow, IEEE
Principal Architect
ALSTOM EAI
11120 NE 33rd Place
Bellevue, WA, USA
phone 425-739-3485
fax 425-889-1700
jay.britton@tde.alstom.com*

1 ABSTRACT

Operation of large-scale synchronous interconnections requires sophisticated computing systems that perform analysis of both what is happening and what might happen on the network. Without such systems, expensive over-building of plant would be required to achieve equivalent reliability. The effect of adding markets into the picture is generally to push even closer to limits and to broaden the impact of network analysis because the financial interests of market participants are affected. All this means that an accurate model of the power system is a fundamental requirement. It is a significant challenge to put together a model for a new operation, but without it, everything else is built on quicksand. Despite these facts, model building processes and tools usually receive secondary engineering attention, and there is currently no well-defined measure or well-accepted standard for model quality.

Achieving an accurate model is laborious and error-prone. In this paper we will discuss two distinct processes which must be carefully designed in order to produce a good result. We will also suggest how model quality can be measured and used to drive model improvement. The first process is initial population of the model – this is a process that is hard to standardize, operates against tight time deadlines, but is a one-time effort. The second process is maintenance / improvement of the model – this is an ongoing process which needs to be fully supported by both business processes and production software.

2 OVERVIEW OF NETWORK MODELING

There are a family of analytical applications which are used in the operation of a large interconnected power system. These include Energy Management System (EMS) applications, such as power flow, state estimation, contingency analysis, and unit commitment, and newer market system applications, such as locational marginal price (LMP), and security constrained unit commitment (SCUC). But we are concerned here with the fact that these applications will all share a common “model” of the network, not

with the specific kinds of network applications. The model describes the connectivity and properties of elements of the electrical system. The model must be shared because the analytical functions feed one another.

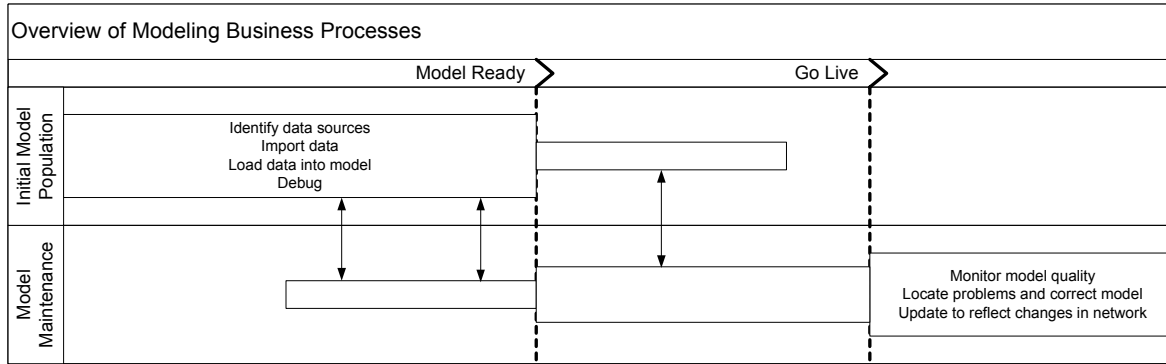
Because it is shared, the scope of the model must include a superset of all the required information for each kind of analysis. This means that the parameters required for each element will be the union of parameters required by any of the applications. It also, though, means that the coverage of the model will be dictated by the union of all needs. The coverage issue includes both territorial extent and degree of detail.

The modeling process is a challenging prerequisite for any operations system. It includes selecting the appropriate scope for the model, locating sources for all the data required, preparing the data in a usable format, debugging the model, installing it, and then finally the initiation of an operations phase of monitoring / maintaining / improving the quality of the model.

One of the most common project management problems for operations systems is underestimating the amount of work which model development requires. This may manifest itself as any combination of 1) not enough time in the schedule, 2) insufficient staff, 3) too little attention to organizing the process and providing the right tools. The consequence of shortcomings in the model is that the analytical functions in the system are either unavailable or cannot be counted on to function properly. In the more traditional operations systems (prior to competitive markets), such unreliability could be tolerated for a period of time, so long as basic functions like SCADA and AGC were operational. Indeed, many systems which initially purchased functions like state estimation and contingency analysis never really achieved full operational effectiveness in these areas (presumably because when it came right down to it, the labor cost of perfecting the modeling was deemed greater than the benefits).

More and more, though, such a result would be unsatisfactory, and with the advent of market systems, calculations such as LMP are at the heart of the financial result and reliable network analysis is an absolutely essential component of a successful system operation. Project management therefore must appreciate both the challenge of modeling and the necessity of modeling. The heart of this challenge is a network model which supports high quality state estimation, and that is the primary focus of this paper.

Project planning should distinguish between the process for initial population of a model and the process for maintaining the model on an ongoing basis. The objective of population is to get a model into good enough condition so that it can support integrated testing of the system ("Model Ready" in the figure) and then commissioning of the system ("Go Live"). Ideally, the population process is complete at Model Ready, but commonly some activity will spill over into the time between Model Ready and Go Live. The criteria for judging whether a model is "good enough" is subjective, but in practice, it usually means that all the analysis functions will provide usable solutions most of the time. There will be hard dates for both integration and commissioning, so initial population works against serious time constraints. In practice, there will be a lot of pressure to go ahead, even though there will be known model quality issues. Population can be treated as a one-time effort, and usually should be viewed this way, in order to minimize time.



The objectives of the maintenance process are to improve the model quality and update the model as changes are made to the network. It is important to realize that most models go live at a level of quality which one would not want to live with forever, so model improvement is an immediate need. The main distinction we are drawing between the population and maintenance processes is that population is a one-time effort while maintenance is ongoing. This means that population tools and techniques can be crude, and manual steps are often cost effective compared with perfecting software. Maintenance tools and techniques, on the other hand, need to be refined to the same levels as any other production systems. Because the distinction we draw is on the approach to process design, rather than strictly a time phase, the two processes overlap in terms of time. Initial population is the only activity at the outset, but portions of the maintenance process become valuable as soon as we start debugging and correcting the model, which happens long before the population processes are complete.

3 INITIAL MODEL POPULATION

Model “population” refers to the initial entry of data into a modeling database.

3.1 One-Time Process

It usually takes quite a bit of work to populate a model, which means that the process should be carefully planned. The first thing to think about is whether the same process will be used for making updates to the model after it is in place. If that is the case, then a repeatable production process is required for building a whole model from original data sources. If it is not the case – if the process is considered as a “one-time” event – then tools and techniques do not have to be as refined.

Keep in mind that a model cannot remain constant:

- The power system undergoes changes over time, and the model must change to keep pace.
- Errors in the model will be discovered, which need to be corrected.

Should these updates be handled by re-building from corrected sources, or should they be made directly to the installed model? It is next to impossible to design a single “best practice” for model population, because systems vary in requirements -- the number of data sources varies, the quality of the data available varies, and so on. However, it is usually very difficult to reliably alter the original data sources and replicate the full re-

build process. Amending the model directly is preferred, which has the added advantage that it means that initial population can be treated as a one-time event, which is usually much less expensive.

- One-time processes can utilize software tools (such as conversion tools) which are not perfect. If they do a 98% job, modeling staff can fix the remaining problems by hand.
- If special software tools are needed, they do not have to be production grade. This means development costs for software can be on the order of 75% lower.
- Some tools development (e.g. use of spreadsheets) may be done casually by the model engineers rather than involving programmers.
- Formal training, formal acceptance are unnecessary.

One-time processes do, however, still require careful planning, and although crude tools like spreadsheets are often very effective for manipulating data, one needs to be careful about designing the work process so that stages of work are backed up and so that multiple workers can be active without coming into conflict.

3.2 Setting the Scope

Setting the scope is a key first step in the process. How much of the network should be modeled? (What regional extent? What voltage levels? How much detail?)

Typically, a given operations center has responsibility for only a subset of the entire territory covered by the synchronous interconnected network. It also may share a list of responsibilities with other centers in the same territory. (e.g. a market system separate from an operations system.) To determine scope,

- Start with the territory covered.
- Look at the functions to determine the level of detail required. For example, you cannot use a measurement in a state estimator unless the component which is measured is modeled accurately.
- Look at the functions to determine how far outside the area of responsibility you need to model. For example, any external element whose outage would create significant impact within the area of responsibility would usually need to be represented.

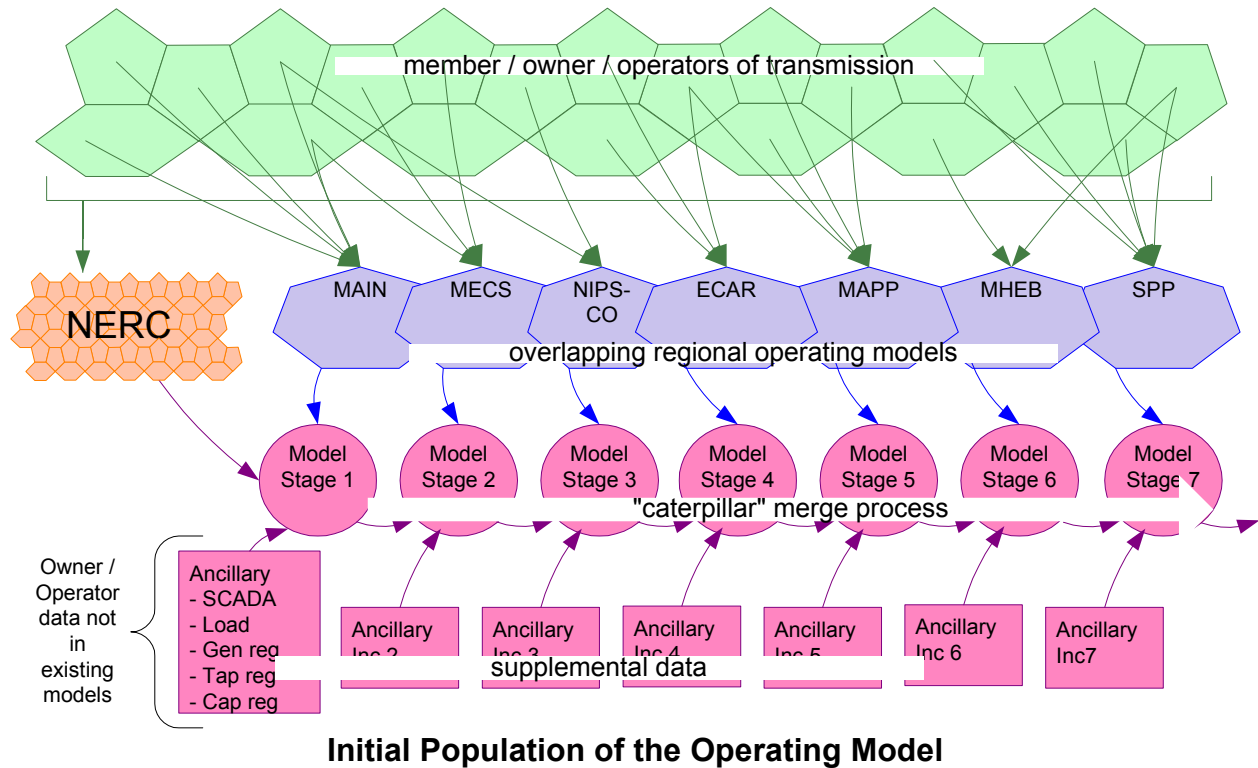
These steps will determine rules for minimum scope. The other consideration which is important is whether the model will go beyond this for reasons of convenience. (It is often easier to take in complete existing models than it is to build new models exactly tailored to the new purpose. This issue is discussed a bit more in the next two sections.)

3.3 Identifying Data Sources

Where does the model data come from? As a rule, it comes from sources outside the operations system itself – often from entirely separate organizations. The most extreme situation is a market system or pool operation center, where virtually all of the data comes from other organizations, and there are many sources.

As part of the basic planning of model population, one needs to identify what data will come from what sources. This will have two main aspects:

- Regional. A given area of the model will come from one source; other areas from other sources.
- Functional. A given source may have only part of the data – for example, bus-branch modeling, so that switching detail has to be added by some other means.

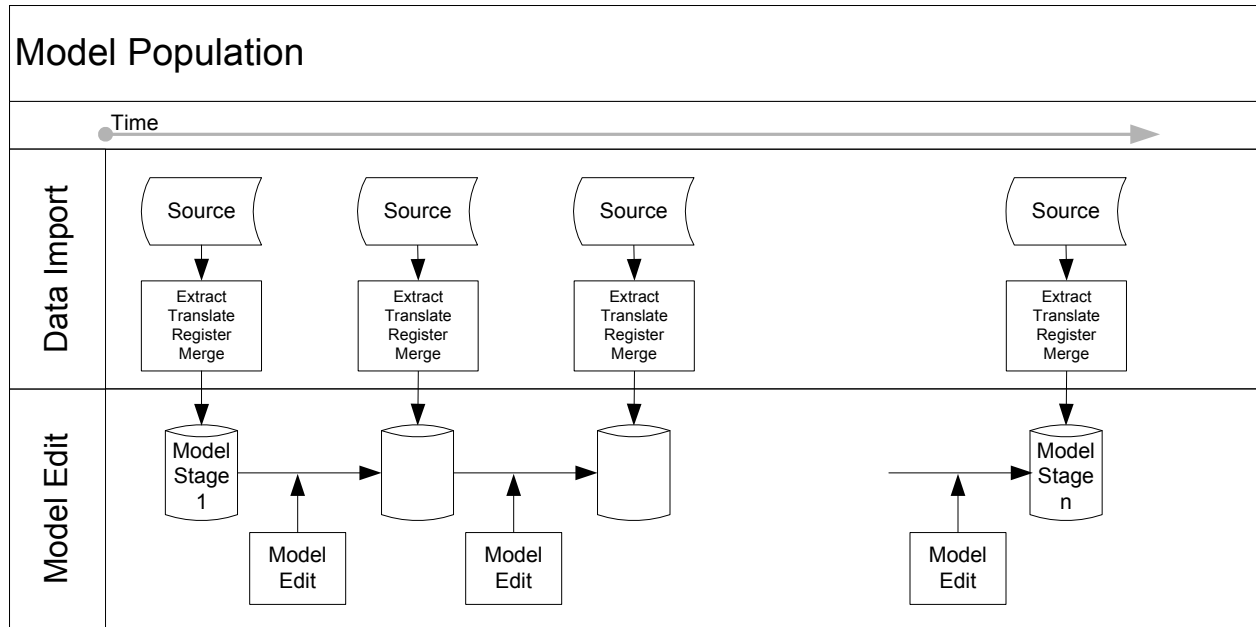


When many sources are involved, there may be corresponding stages of development and checkout. This might result in a conceptualization of the population plan such as the one shown above, which was created for a North American ISO. It involved initially staging a planning model (from NERC) which had broad coverage but lacked detail, and then successively extracting detail from other regional sources and merging that detail into the broader model, using power flow at each stage to check results before moving on to the next stage. We called this a “caterpillar” process because of the segmentation.

3.4 Data Import

Data import is occasionally all manual (in which case data entry labor is the straightforward answer as the import process), but more often, some sort of computer database or file exists as a possible source, and we want to automate the following:

- Extraction of the portion of the source data that is of interest.
- Transformation of the data from source structure to the model structure.
- Registration of instance names and rationalization with model instances.
- Merge of new data into the model, deleting duplications and matching up relationships between new and old parts.



Sometimes complete automation is impossible (because the source data is incomplete). More often, complete automation is simply impractical. Always, manual edits are required. The tradeoff is how much to do with software and how much to do manually. In theory, manual corrections may be done at the source and re-imported, but this is usually not very workable. In practice, data import proceeds as shown in the diagram. Automated import brings in a set of data, which is then edited on the model side. As edits are done on the model side, it becomes increasingly undesirable to have to go back and repeat the import. Saving the model at each stage of development is critical so that when some data disaster happens, the process need only back up to a point just before the mistakes were made. Saving the edits is also very valuable. (This will be discussed more as part of the Maintenance Process.)

Sometimes software for an import exists. This is frequently true when connecting between major vendors or where one of the formats is the CIM (see next section). At other times, software will have to be customized or developed from scratch. Two things are important to know:

- Even existing commercial software for imports is rarely perfect. You will still need to fix the results manually.
- Developing effective imports can be relatively inexpensive, provided it is treated as one-time, throw-away code, and provided that it doesn't have to be perfect. Costs go up by an order of magnitude when production-grade standards are applied.

3.5 CIM Standard

Transforming information structure is an obstacle because the network model is quite complex and different developers of products over the years have used many variations. In recent years, there has been an industry effort to develop a Common Information Model (CIM), in order to have one information structure that could resolve the differences. Significant portions of the CIM have been adopted by the IEC as industry standards. (The reader is referred to the IEC 61970 series of documents which has been

produced by WG13 of TC57.) The greatest contribution of CIM is toward the creation of an overarching model which has enough richness to supply all of the applications – this is no small design job.

The main effect of CIM on the data import problem is that it can be used to assemble the model. In other words, everything is translated into a CIM-compliant structure. All the import tools produce CIM-compliant results. All the management tools work are CIM-compliant. As more CIM-based tools are developed over time, this will become a bigger and bigger advantage.

The CIM effort to date has not attempted to look at the overall processes for modeling, nor the functionality which modeling tools should have, so one should not equate “CIM-based” with any particular level of tools functionality. In most cases, a phrase like “CIM-based” won’t necessarily even mean that all CIM structures are understood. It will only mean that what is implemented by the tool, is implemented using CIM structures.

3.6 Naming Conventions

If one was building a model entirely from scratch, one would first do a little thinking about how to identify all the elements, and a naming convention would be the result. The naming convention would satisfy the user requirement to be able to recognize the name, and it would satisfy the computer requirement of providing a unique tag for each item.

In reality, names always exist already. Often, a given network element has more than one name, a result of being assigned names independently by different parties that have modeled it before. Similarly, different parties may have coincidentally used the same name to refer to two different things. When information from separate sources is combined in the process of creating a model (which is very common), then both of these kinds of naming conflicts always become a problem.

The immediate temptation with names is to try to find a new “globally unique” naming convention (as if this was a new model from scratch). There are problems with this:

- First, there is no such thing as “globally unique”. There is only “unique within the scope of those who play by the same rules”. Invariably, adherents to a naming practice will run into the need to exchange information with systems which are already in place with non-conforming naming conventions. You just can’t get everybody on the same page at the same time.
- Second, names must communicate with humans. Humans in different roles or different organizations may have very good reasons for using different names.

In the end, naming conventions and unique identifiers are valuable, but so is a systematic way to manage naming correspondences. A name registry should be employed for this purpose and the modeling team should understand how to register names and manage correspondences. This is a relatively new idea which is just now seeing some recognition in standards groups (NERC DEWG and IEC TC57, WG13-14) and in product development, but with the increasing importance of integration of computing systems, this is an important area to pay attention to.

3.7 Comments on Functional Subsets of the Model

The overall model has a number of semi-distinct information areas.

3.7.1 Network Physical Equipment

The most basic set of information is that which models the physical plant. In an electrical network, in which the electrical connectivity of each element of plant to other elements must be carefully defined, equipment becomes more densely interrelated than in many other industrial settings. The network is really one huge machine, with an enormous number of component parts, but with oddities such as widely distributed geography, ownership and operational responsibility.

- The easiest sources for data are existing models, but keep in mind that all data has errors in it – even data which has been used a lot.
- The initial difficulty will be the piecing together of models, because the boundaries won't match without some repair.
- The ongoing difficulty will be that the detail in the source does not exactly match with the new purpose.

3.7.2 Setting up Real-time Measurements

Most operational systems use a state estimator to find the best fit of real-time measurements to the model. A fundamental part of the static model specifies precisely what, in the network model, is being measured by each measurement. Several issues arise:

- A modern estimator will provide a solution even if there are no measurements, but the accuracy is no greater than for a study power flow. To get statistical improvement of fit, there must be “redundancy”, which means that a state estimation specialist should review the telemetry available and make recommendations as to what telemetry is required in order to fulfill the operational purpose of the system.
- Existence of telemetry is a main driver of the detail required in the model. Any device which is measured must be explicitly and accurately represented.
- A “mapping” needs to be established between each telemetry item and the network point it measures. This is usually a manual and error-prone process, though occasionally one can find ways to automate.

3.7.3 Loads and Other Equivalents

Models always consist of physical modeling (where real equipment shows up explicitly in the model) and equivalent representations (where portions of the power system are simplified into electrical approximations and the identity of real equipment is lost). The most common equivalents are loads, which replace the need for modeling of lower voltages. Network equivalents are also fairly common, in which a larger real network (usually outside the operation center's responsibility) is simplified to a smaller number of equivalent nodes and branches. Both types of equivalents cause real headaches.

Loads could all be real, if we modeled down to all customer meters. Usually we don't do this. We cut off modeling at lower voltages so that transformers or feeders become aggregated loads. Data about how load forecasts should be distributed to these load points is among the least available and least reliable information we have, although it can be tuned up through state estimation as the system goes into operation. We wind up depending on state estimation here for accurate real-time information, rather than the load

models, so this should be a consideration when the availability of measurements is being considered.

Network equivalents are even more problematic, though less important. They will mathematically match an operating point, but are less reliable over a whole range of conditions. They are a nuisance, because they are imaginary, so that names and values do not make any intuitive sense. They have to be regenerated regularly and replaced as a whole, similar to merging a new sub-model. This raises the question of how the source model is kept up to date and in synch with the operations model which is being maintained. If you have both models available, why not use the detail directly? If not, how do you know whether the equivalent is valid? The reliable way to use equivalents is to maintain the full model in detail and derive equivalents yourself for use in network analysis – presuming, of course, that you get a reduction in computational load sufficient to justify the extra complication. The best advice about network equivalents is – do not use them.

3.7.4 Market Data

Market data must be added if the operation in question is a market. Much of this information comes from “participants” in the market. A participant must register its generators or loads, for example, prior to being able to do business with the market. This is distinguished from network data mainly in terms of who is responsible for providing the data.

3.8 Debugging to the Point of Feasibility

Model “feasibility” is not a precise concept, but roughly speaking, a model is feasible for a given analytical function when its algorithm will converge to a usable solution most of the time, and it is generally feasible (i.e. it could go live) when all analytical functions are feasible. Models are virtually never feasible when the initial import is done.

In the model population process, debugging to achieve power flow feasibility is the basic step which is repeated after each new major stage of additions is received, and then general feasibility is pursued for the whole model. Debugging involves detecting and correcting problems. The correction mechanisms are basically all the same as for maintenance activity, which will be discussed later, but the detection problem has a different character while the model is infeasible.

Infeasibility usually results because the data is grossly wrong. More than likely, the data depicts a situation which is physically impossible, which means in turn that the algorithm can’t find a solution (because one does not exist). Finding the problem can be difficult when there are no meaningful solutions to look at and diagnose. However, experienced engineers can use screening tools to look for common problems (such as unrealistic R/X ratios), and eventually, the data will support a solution of some sort. Usually, one wants to start with an intermediate case (like an 80% of peak load case), to make the process as easy as possible. We want to concentrate first on whether the physical network model is reasonable. Then come back around later to whether the load models will cover the full range of conditions in a reasonable way.

When a power flow case begins solving, this means that the data describes a physically realizable system (at least in terms of Kirchhoff’s Laws), and the result can be examined for unlikely conditions (e.g. very low voltages) which would indicate poor areas of modeling. The “worst mismatch” locations are also good clues that there is some bad data

in the area. Power flow results can be used to generate “perfect” measurements to be used in trying state estimator solutions.

Many EMS systems have gone live without achieving solving models, but this is far from ideal, and it is unacceptable in a market system. The minimum requirement for a market system is that the state estimator and power flow both find solutions reliably if the realtime data is accurate. In other words, you may go live with some questions about the fidelity of the model to the physical system, but it will reliably find a solution. This is the significant milestone for the modeling team – the point at which the main concern switches from “does it solve?” to “how accurate is it?”.

4 MODEL MAINTENANCE

Model Maintenance refers to the processes for making changes to a model, either to correct problems or to add modeling that reflects changes to the power system. The sections which follow first discuss the basics of making changes, and then talk about the process which should be set up to identify the changes which need to be made.

4.1 Cases and Changes

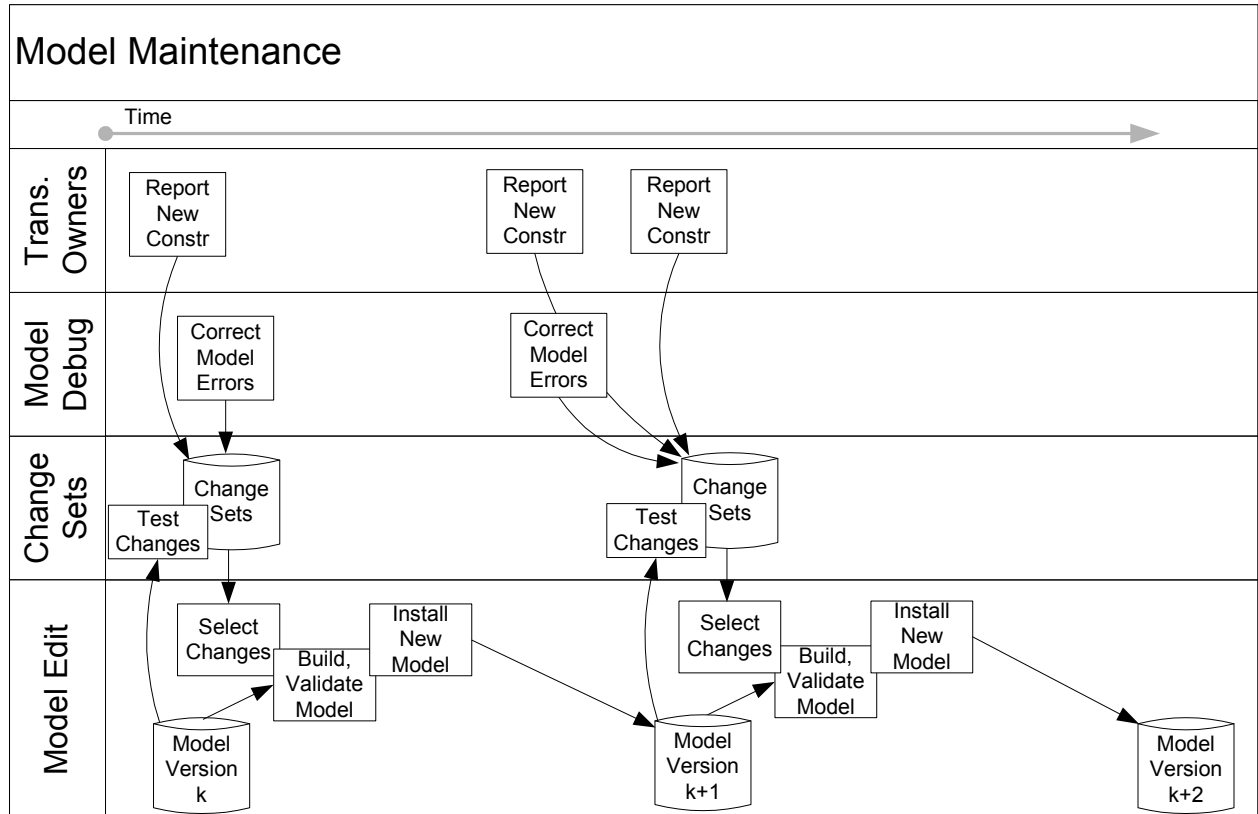
In an earlier figure, we showed Model Population as a sequence of model stages of development, and we discussed the need to save these stages. Let’s define a saved copy of a complete model as a “case”. Just as model population involves a series of cases, so does model maintenance. Each maintenance stage consists of a set of modifications, and results in the installation of a new whole model. (The diagram looks identical on the surface and the tools for doing maintenance are useful during population as well.) It is necessary to retain saved copies of the latest “on-line” model, which might also be called the “as-built” model of the system. It is wise also to retain recent previous models, so that one can retreat to earlier models if some sort of data disaster occurs. In some contexts (such as markets) there may even be an auditing requirement to save all models used, carefully documented in terms of which were used during each time period.

Bottom line: there will be a directory system for archiving model “cases”. This is straightforward to accomplish, though it may use up a lot of space.

The tools for making changes and managing change data is where most systems are deficient. Common practice is to load a case into an editing environment, such as a database, then make changes, then save the resulting case. Though this is workable, there are problems:

- There is no good connection back to the source of the changes. Some changes, such as new construction, originate outside the system and a systematic way of reporting the changes is desirable. Other changes may originate inside as a result of model debugging.
- A given new case often represents several separate functional changes, but the separate history of these usually cannot easily be tracked.
- The common vehicles for creating audit trails of changes are database journals or difference files (the result of comparing two cases). Neither of these makes very readable documentation of what the changes are.

New proposal: What is needed is a separate data structure for describing a change – call these “change sets”. Change sets are something like “work orders” for the model, where the idea is to create separate descriptions of each functional grouping of changes (representing one new construction project, or the correction of a particular error, for example). Change sets would include documentation (name, person responsible, organization, descriptive text) as well as the actual model changes. If data sources outside the operations system are supposed to report changes, they might use a web interface to create and submit a change set. If a change is generated from a program (like a power flow) it should be able to produce a change set. This allows a much more fine-grained management of new model information.



With change sets, the process for change management looks a little different. As problems are identified or new work is reported, each is individually described in a change set. Each can be validated by itself with testing to see how the existing model would behave if the new change were installed. Then, as it becomes desirable to install changes, the desired sets of changes can be combined into a new model version and put on-line. This sort of procedure is more flexible and much better organized. A clear documentation trail can be maintained. Maintenance work is more easily divided among personnel. Planned activities can be maintained separate from current modeling, and conveniently studied in advance. And as a bonus, the need for full saved models should be reduced, which will save space.

4.2 Model Update

One cause of model maintenance activity is that changes have occurred in the power system, and new modeling is required to reflect those changes. The changes could be new construction activity, new load models, new operational constraints (such as ratings or voltage ranges). There is nothing wrong with the existing model – it is just going to get out of date if it is not updated. The party which initiates the change is often a separate organizational entity. (In market systems, all such changes usually come from other corporations.)

Model updates primarily pose an organizational challenge – though one might also add that existing software systems are often short on effective support. The first step in accomplishing effective model update is to sit down with the sources of information and agree on the update procedures and the responsibility of the parties involved. For example, the table below outlines basic steps in a process for reporting new construction.

Step	Functional Step	Mechanism
1.	New construction activity is identified by the transmission owner in advance of the change.	Transmission owner uses web UI to create and submit a change set that describes the planned change.
2.	The planned change is queued for processing by the model maintenance personnel.	A work tracking system is useful to record the change set submitted and subsequent process. (This could be just a spreadsheet, unless more rigorous tracking is required.)
3.	Model update data is checked out for numerical problems in advance by running power flow studies. If problems are found they are fed back to the source.	A temporary case is set up of the current model plus the submitted change set. Power flow is used in study mode to review the result.
4.	Change is installed prior to completion of new construction.	Change set is merged with other changes to build a new model version. Open switches typically remove the change from any real effect until the change goes live.
5.	Energization of new construction is monitored by maintenance personnel to see how model behaves.	Switches are closed as construction is energized. State estimator results are reviewed. Significant residuals in the changed area indicate the need for model review.

Note that the above depends on change sets as the medium for managing this process. In most current installations, the best one could do for a change set is a word document, which, of course, would require manual transfer into any cases in order to be put into effect. The CIM, however, gives us a very good basis for encoding changes and implementing automated processing – and this is how future systems should be designed.

4.3 Model Improvement

Once a model is basically working, one might ask, “how good is my model?” It is remarkable that this question has not been asked very often in the history of EMS to date, and there has been an embarrassing lack of attention given to this subject in the

engineering literature. Nevertheless, it is important to begin asking and answering this question -- market systems, where the model directly affects prices, are very vulnerable to criticism if they cannot justify their price calculations.

The answer starts with a means of judging how good the model is, and from there implementing a well-organized process of model improvement by continually identifying worst errors and correcting them.

4.3.1 State Estimator and Model Quality

State estimation is the key to model quality because it provides a natural measurement of quality. This fact has not been widely acknowledged or acted upon. Historically, the “justification” of an estimator in the literature has been that it improves (and fills in) the measurements. The fact that non-zero residuals (i.e. differences between measured and estimated values) result if either model or measurements are in error has been largely ignored under the naïve and implicit assumption that modeling errors would be eliminated before an estimator was put in service. Universal experience with project start-ups says this is unlikely to be true. Errors in the static modeling (physical parameters, connectivity, mapping of measurements to model) are always the dominant cause of residuals starting out. The act of getting a system through commissioning gets rid of the most debilitating errors, but there is no evidence of which we are aware that model errors do not still dominate as the source of residuals. And it is very clear that major payoff of state estimation is that it points out areas of modeling which must be cleaned up.

This lesson is usually not converted into a formal process for model quality and improvement. We propose that it should be. Errors in models yield errors in calculations under the venerable principle of “garbage in → garbage out”. When the result quality is important, the model quality must be equally important.

The state estimator objective function (which measures fit) would be one candidate for measuring quality, but this is typically contaminated by weighting factors which are good for the numerical process, but may obscure the quality issue. We propose that there should be an independent gauge of quality and that it should be tuned to the purpose of the system. For example, if we are dealing with an ISO market system, the value of accurate MW representation within the footprint of the ISO would generally far outweigh the value of accurate reactive and accurate external modeling, though there might be some exceptions – for example, in cases where a system was prone to voltage collapse. We think that each system should tune their own specific criteria, but a good starting position for a market system would be:

- Measure quality of every solution by producing lists of the “n worst discrepancies” in a given class. For example, “the 10 worst MW differences within the market footprint”.
- Track progress by trending. For example, trend “the sum of absolute MW differences in the market footprint”.

4.3.2 Driving a Model Quality Improvement Process – Problem Detection

The first purpose of model quality indication is to measure how well we are doing, but this is of comparatively little value unless it is teamed up with a process for improving quality. And it is very important to engineer this process.

Here's the challenge. Theoretically, we know that residuals are caused by either measurement or modeling errors of varying classes. How, though, can we distinguish which kind of problem is the source of a given residual? The answer has defied software engineers who assumed that the answer to such "parameter estimation" had to be algorithmic, but operating engineers have demonstrated over and over that they can find the source of a given problem with a little persistence and intuition. The practical answer is to engineer a process that incorporates the human intelligence efficiently, rather than bypass this valuable resource. The goal should be to feed the operations personnel information in the most useful form – then let them apply their intuition – then give them convenient tools for fixing the problem.

So, we need to look at what might be going wrong and what kinds of symptoms would be produced. First, there are three major classes of data which can be responsible for state estimator residuals:

- Static modeling errors.
- Analog measurement errors.
- Supplemental manually entered real-time data. (Unmetered switch positions, tap positions, plus pseudo-measurements produced from voltage schedules, load / generation / interchange schedules.)

Second, we can easily distinguish the following very useful information about residuals:

- What residuals are significantly higher than others? Especially, what are the worst residuals and how much worse are they than the average?
- Of these worst residuals, have they been there consistently? or did they just appear in this latest state estimation? If they just appeared, what changed?

Presenting the above information in a well-organized way provides very powerful clues with which to find the cause. If a particular measurement residual is consistently high, it might be that the physical measurement is badly calibrated, but it is more likely that some part of the static modeling of either the measurement or the network model (near the measurement) is wrong. Electrical proximity is the primary clue for finding the problem. Most problems are not hard to see once you are pointed to the right location – a few are difficult. Usually, this is an off-line task performed by administrative staff rather than power systems operations personnel. They need to have the problem areas reported and queued for their attention. And they need to be able to get the real-time cases (with measurements) for use in testing hypotheses. This means that software for archiving all the inputs to the state estimation is important to enabling the detective work.

If a problem has just appeared, then clearly something must have just happened which will provide a clue. The most essential problem detection aid after the quality presentation is a summary of what significant changes have occurred in the input data. What we really want to screen out is:

- Were there significant changes in the list of worst residuals?
- What input changes occurred which might be responsible?

When a state estimator suddenly stops solving, or when new large residuals occur, the immediate area of suspicion is that the supplemental real-time data is out of date. Here are some examples of the reasoning that would take place:

- If a significant error appears and there was no change in input which could be responsible, then the likely situation is that there has been a change in a real device which has not yet been reflected in the supplemental data (probably a switch position).
- If a significant error appears and there were input changes, then either the new input was wrong, or the new input exposed an existing modeling problem. Both are common. For example, telemetry may give false input by report changes on a breaker that has been isolated by untelemetered or unrepresented switching. Or, a real switch status change may reveal incorrect topology modeling.

Algorithmic deduction of problem from symptom is not generally possible (or at least, not cost effective), but most problems are readily identifiable if the right data is presented and if the operation dedicates staff to maintaining quality. This latter is important so that the staff develop the necessary intuition.

4.3.3 Fixing the Model -- Problem Correction

Having identified a problem, how do we fix it? Of course this depends on what sort of problem it is, and there is a very broad range of possibilities, given the diverse set of input. However, we can boil this down by looking at some general types of problems and thinking about how to organize people to deal with the problems. There are really two major business processes which should be considered.

- The Real-Time Model Process deals corrections which can be made in the short run, which tends to be mainly managing the supplemental data.
- The Static Model Process deals with problems which take longer to analyze and correct, primarily dealing with the static model.

4.3.3.1 Real-Time Model Correction

An operations system should have operations support personnel identified whose job description includes maintaining the accuracy of the real-time data. Their objective is literally to minimize the area under the spikes in the quality trends. In other words, when quality suddenly deteriorates, find out what the source is and fix it as soon as possible, because the financial consequences are piling up in proportion to the size of the error times the length of time the error remains. If the operation is critical, this job may need to be manned 24x7.

Problems are very simple to fix if the operations personnel are already aware of roughly what is going on. For example, if it is known that there is construction going on in a given substation, it is probably fairly easy to deduce what sort of correction is needed, even without direct communication with the people working in the field.

At other times, however, the time to correct a problem may have to include time to communicate with another organizational entity. Part of the process of developing a smooth-running organization will be establishing procedures with the other organizations for responding to inquiries – or perhaps reporting ahead of time what events are going on which will affect the state estimation process.

4.3.3.2 Static Model Correction

An operations system should plan to have a support staff whose job is to maintain the static model. (Warning: it is common to underestimate the staff required, but there is no good rule because the requirement varies with the size of the network, the quality of the original data, the difficulty in dealing with sources, and the need for quality.) The objective is to improve the average quality continually – in addition to handling the model updates that reflect changes to the power system over time. If one is tracking quality, the trend of average quality over time should be a good measure of their success.

Improving average quality involves continual repetition of the following steps:

- Take the worst recurring residual and define a new work task for fixing it.
- Set up a study case from captured real-time data.
- Develop a hypothesis about what is causing the problem.
- Communicate with the data source to verify the hypothesis. (e.g. If you think a tap is in the wrong position, ask the transmission owner to check what position the tap is in, and how the tap is normally operated.) Go back to the previous step if your hypothesis cannot be verified.
- When you have the answer, complete the change set describing the solution.
- Schedule the change into a new model version.

4.3.3.3 Crossovers

Naturally, problems don't always fall cleanly into either real-time or static classifications, so communication between the two processes is important. Sometimes, a quality spike happens because a change in the system exposes a static modeling problem. Sometimes, static modeling research identifies problems which can be temporarily corrected via real-time data, such as removal of measurements from service, or override of measured status.

All maintenance activity should be logged and tracked within one work management system to help keep both processes alert to activity in the other.

4.4 How Good Can It Get?

There is no clear answer to this question, since we do not have a base of real experience with quality management processes.

Theoretically, if the measurements were all accurate, and we eliminated all modeling errors, all residuals would be zero. In practice, factors like transducer accuracy, time skew in reporting measurements, and calibration mean there is always some level of analog measurement error. Modeling error adds to this base. No good data exists on anything other than the ideal accuracy of metering equipment itself.

Currently, these practical suggestions can be made:

- Monitor quality improvement in terms of quality spikes and average quality. Leave the processes fully staffed until progress flattens out.
- Look at quality in terms of how much real impact the worst error has.
- Compare quality measures with other installations.

5 CONCLUSION

In the design, construction and execution of operation systems for electrical systems, it is pretty universal that the subject of network modeling and model quality takes a secondary position to the business of defining on-line functions. Yet it is also true that the success and accuracy of these on-line functions always depends on the accuracy of the models, and many projects have been delayed by underestimating the time required to get models into feasible condition, or have wished belatedly that they had paid more attention to tools for efficient maintenance of models.

In this paper, we have provided an overview of modeling and made many suggestions for organizing the process and improving the tools which support the process. We would like to highlight the following in conclusion:

- Model quality is important. (Garbage in, garbage out.)
- Initial creation of a model is different from maintenance, and should be treated as a one-time event.
- Getting to the original data can be time-consuming, but trusting intermediate sources is risky.
- Name translations should be managed with a name registry. Don't depend on universal naming conventions.
- Assume that the initial model will be of inadequate quality for sustained use. Plan on continual model quality monitoring and improvement.
- The state estimator is the key to quality monitoring. Make sure it provides feedback to the quality improvement process.
- Good maintenance tools will improve efficiency dramatically and provide audit trails which document performance. This has been a neglected area – look carefully at software offerings in this area.

6 REFERENCES

Standards:

- [1] *Draft IEC 61970: Energy Management System Application Programming Interface (EMS-API) – Part 1: Guidelines and General Requirements*, Draft Revision 7.2, Feb., 2003.
- [2] *Draft IEC 61970: Energy Management System Application Programming Interface (EMS-API) – Part 301: Common Information Model*, Revision 6, Feb., 2002.