

Defining a Model: A Vision for a PowerWorld Case *Builder*



Jamie Weber, Ph.D.

Director of Software Development

weber@powerworld.com



PowerWorld
Corporation

2001 South First Street
Champaign, Illinois 61820
+1 (217) 384.6330

support@powerworld.com
<http://www.powerworld.com>

Vision of *Builder*



- Define Base Data Structure
 - Define a set of relational database tables the fully define what you need in the base system topology
- Define method for structural modifications
- Define method for discrete time modifications
- Define method for continuously time-varying modifications
- Defining related data (Contingencies, RAS, Geography, Onelines, etc.)
- How to access the data to provide updates and download cases at a particular time?
- How to ensure the cases created by this vision reliably solve

Vision of *Builder*

Step #1: Base Data Structure



- Base Data Structure
 - Define a set of relational database tables that fully define what you need in the base system topology
 - For testing purposes, this can be encoded in a PowerWorld AUX file → ultimately it would likely reside in a SQL server environment
 - What to keep/throw-out
 - We do not need “ProjID”, “Start Date”, “Retirement Date”. These fields in an EPC file really don’t make sense in this structure.
 - Keep multi-section line structure or throw it out?
 - Replace with something more flexible and consistent
 - Document in detail how this structure maps to a RAW file and an EPC file
 - This could be used by Columbia Grid in their existing case building process amongst members

Vision of *Builder*

Step #2: Structural Modifications



- Define method for pushing in structural modifications to the base structure
 - Would include an effective date/time of the change
 - Provide ability to add new objects and delete existing objects
 - Discussion of whether ability to move an existing object is desired (move a generator to a new terminal bus for instance). Other choice is to treat movement as a deletion and an addition
 - Build software tool to generate this
 - This would build on top of Simulator's Present Topological Changes from Present Case Tool

Vision of *Builder*

Step #3: Discrete Time Modifications



- Define method for pushing in discrete time modifications
 - Again this would include an effective data/time of the change
 - Examples
 - Seasonal MVA Limits
 - Seasonal Generator MW limits
 - How should we identify these objects?
 - Might be wise to use to use labels here. Especially if we permit folks to move terminal buses of various objects
 - Again, we would discuss
 - Syntax would likely be very simple
 - Date/Time, ObjectString, [VariableNameList], [NewValueList]

Vision of *Builder*

Step #4: Continuously time-varying



- Define method for pushing in continuously time-varying values
 - MW/Mvar Load schedules (EMS systems do this using something that looks like an Injection Group)
 - Would this represent the load at 3 PM each day for example?
 - Do we want this to handle hourly variation inside of a day → maybe have one time-varying multiplier for daily peak and a second time varying multiplier over the course of a day?
 - Scheduled Area-To-Area Transactions
 - Generator MW/Mvar schedules → wind/solar generation (uncontrolled)
 - Generator, Transformer, Shunt Voltage Schedules?
 - Bus-based voltage schedules?
 - Other?
 - All of these structures would likely include a time-varying piece-wise linear lookup associated with it

Vision of *Builder*

Step **: Defining Related Data



- Defining related data:
 - Contingency and RAS definitions
 - Geographic Information
 - Transmission line routing information
 - One-line Diagrams
 - Substation Layout information

Vision of *Builder*

Step **: How to access this data



- Probably SQL Server
- Also a new PowerWorld Builder Client
- Move to web-based interface
- Data Security
 - At some point in future, it would make sense to have PowerWorld work with another software shop more familiar

Vision of *Builder*

Step **: Ensure Cases Solve



- How to ensure the cases created by this vision reliably solve
 - Build in “check-points” in the time series that the engineer managing the Builder Database would essentially create a new “base case” from that time going forward
 - Potential new solution methods

Enough with Visioning!



How will this work?



- Keep *thinking* about the vision, but stop *talking* about it
 - Do not attempt to write down this entire vision
 - Do not write a 500 entry long RFP table listing all the requirements
- We've started Step #1 working with Columbia Grid in the Northwest.
 - The goal is just to come to an initial draft of what information defines a power system model
 - Obviously this will change over time!

Vision of *Builder*

Step #1: Base Data Structure



- Base Data Structure
 - Define a set of relational database tables that fully define what you need in the base system topology
 - For testing purposes, this can be encoded in a PowerWorld AUX file
 - Ultimately it would likely reside in a SQL server environment
 - What to keep/throw-out
 - We do not need “ProjID”, “Start Date”, “Retirement Date”. These fields in an EPC file really don’t make sense in this structure.
 - Keep multi-section line structure or throw it out?
 - Replace with something more flexible and consistent
 - Document in detail how this structure maps to a RAW file and an EPC file
 - This could be used by Columbia Grid in their existing case building process amongst members

How is this working?



- PowerWorld is working with 3 engineers within Columbia Grid on initial testing
 - We've been talking with them frequently getting feedback a few times a week
- Just talking about this project over the past several months before starting it resulted in changes
 - Motivated addition of the new Data Maintainer structure
 - Motivated additional owners in PowerWorld Simulator

Changes already made



- Modified to allow editing of various shunt and lineshunt fields related to SVCs in Nominal Mvar (instead of in per unit admittance values)
- Completely redid how you manually define three-winding transformers inside the Simulator User Interface dialogs
 - Dialog exists to enter all input data on the transformer base for three-winding transformers now exists
 - Also learned about another RAW → EPC file conversion issue: EPC files only permit variable taps on the primary winding of three-winding

Switched ShuntMode



- Presently options are Fixed, Discrete, Continuous, Bus Shunt, and SVC
 - No equivalent of the “AutoControl” field for a transformer is available
 - In order to turn a continuous shunt off control you need to change it’s ShuntMode to Fixed which means you lose where it started
- Fix we’ll be adding shortly to a patch of Simulator 19
 - Add an explicit AutoControl field for a shunt with choices of
 - NO : means it is not controlled
 - YES : means it is available for control if the Area field AutoControlShunt = YES and the global option to enable shunts to move is enabled
 - FORCE : means it is available for control and it ignores the Area field AutoControlShunt and the global option regarding shunt control (this is an additional wrinkle that folks have been asking for)
 - May also add a new field for CTGAutoControl to specify how shunts behave in the post-contingency power flow solution

LineShunt Object



- Terrible in all software data definitions
 - Doesn't allow automatic control in any data definition (EPC, RAW, PowerWorld AUX).
 - RAW file doesn't support object at all (just fields on a branch for total line shunt)
- Future Fix decided upon (which will be in Version 20)
 - A new field will be added to a switched shunt object that is the "StatusBranch" referencing a branch.
 - If the StatusBranch is OPEN then this shunt will be treated as OPEN automatically.
 - Existing LineShunt object will be removed
 - Will still read/write appropriately to old PowerWorld formats)

Important Notes



- Small project → immediate changes that are useful to users