Electromagnetic Transient Model Interoperability

ORNL EMT Webinar April 21, 2026

Thomas E. McDermott, PhD, PE, LFIEEE

Email: tom@meltran.com or t.mcdermott@ieee.org

Resume: https://meltran.com/files/CV_McDermott.pdf

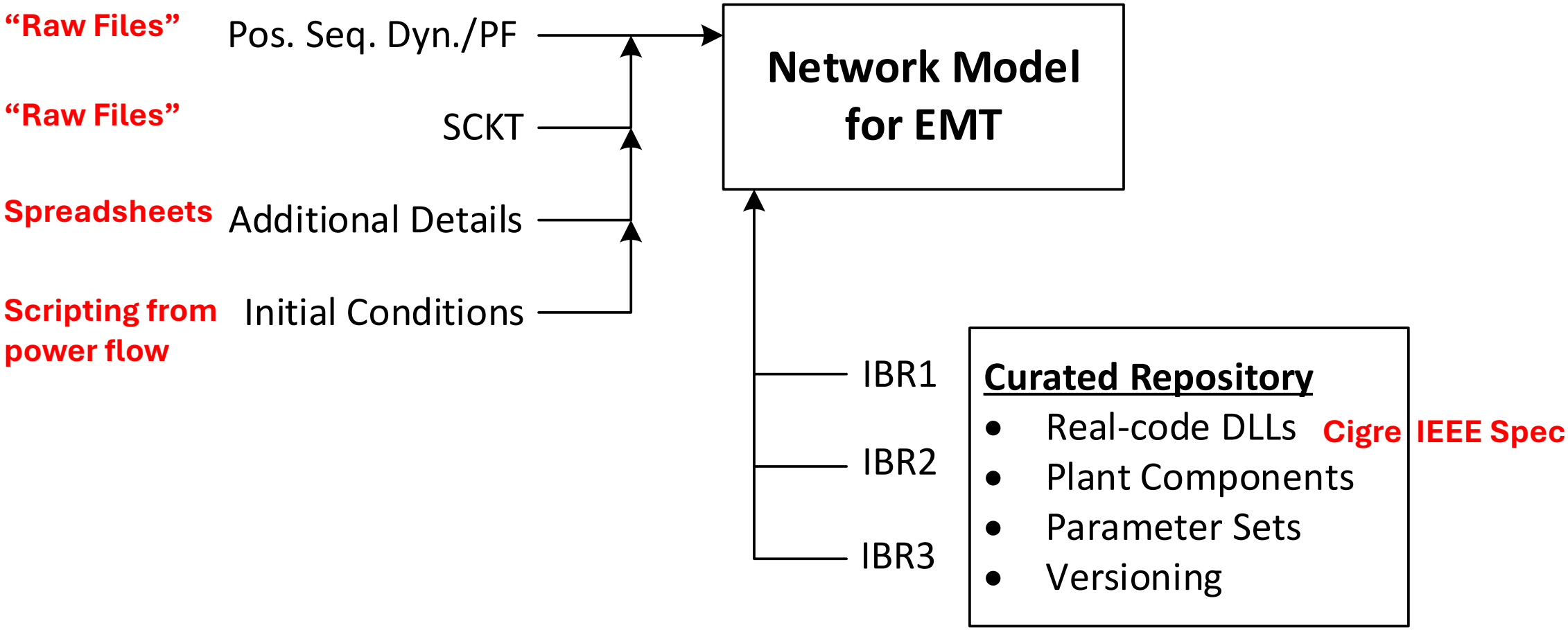
Read the PAR and Sign Up for IEEE P3743:

https://standards.ieee.org/ieee/3743/12233/

Companion Open-Source Software:

https://emthub.readthedocs.io/en/latest/

A NERC white paper identified “attributes of an EMT network and IBR model repository”, which is presently an ad hoc process.



Ref: https://www.nerc.com/comm/RSTCReviewItems/1_08_Draft%20Whitepaper_%20EMT%20Analysis%20in%20Operations_v2.0_clean.pdf

Also See: https://doi.org/10.1109/ACCESS.2023.3305394 presented by Thai-Thanh Nguyen at the March 26, 2024, meeting of the NERC EMT TF.

The Common Information Model (CIM) started around 1990 to support interoperable control center applications and has grown.

Timeline of Some Adoptions and Use Cases:

- ≈1990: EPRI Control Center API Project
- ≈2004: First IEC Std. 61970 Published
- ≈2009: Common Grid Model Exchange Spec. (CGMES) in Europe
- ≈2009: ERCOT Nodal Market uses CIM in Texas
- 2023: NERC Reliability Guideline w/ IBR EMT
- 2025: IEEE P3743, a "Simplified CIM for IBR EMT"

U.S. Dept. of State, Public Domain, 10/5/2022

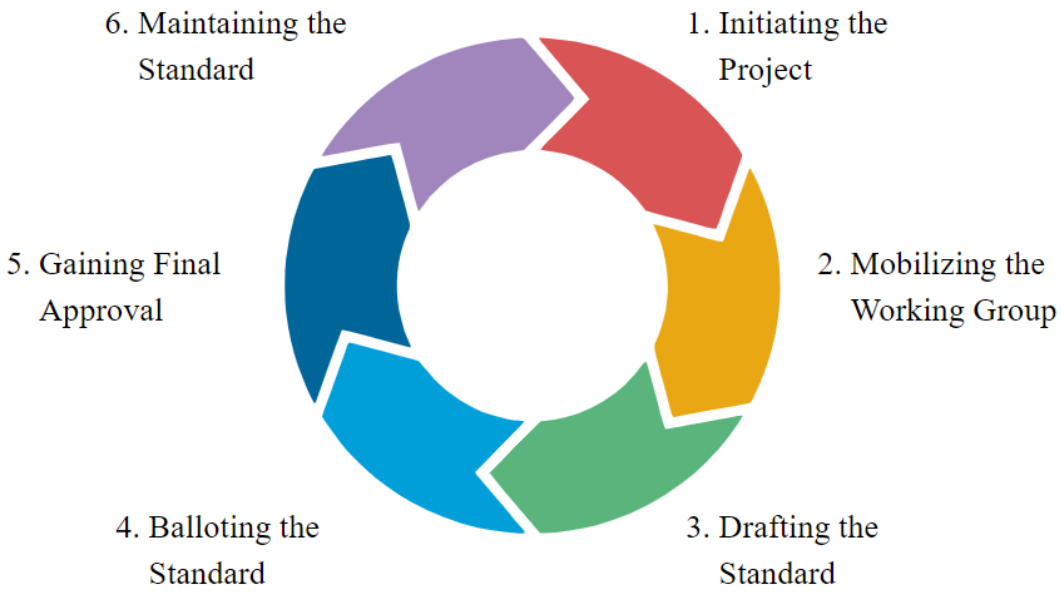
Why not stick with the raw files?

1. It is a vendor-controlled format with multiple versions and EMT gaps.
   a) Duplicative and possibly error-prone reverse engineering by multiple stakeholders.
   b) A standards development organization (SDO) won't adopt a proprietary format.
2. March/April 2025: the NERC EMT Working Group approached the vendor about adding EMT extensions to a newer JSON raw file schema to standardize. This was not successful.

IEEE P3743 began monthly virtual meetings in January. An open in-person review meeting will be held at the PES General Meeting in July.

Revise or Withdraw every 10 years

Project Authorization Request (PAR)



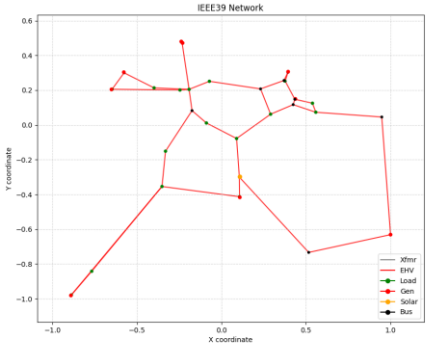
75% returned,
75% approval

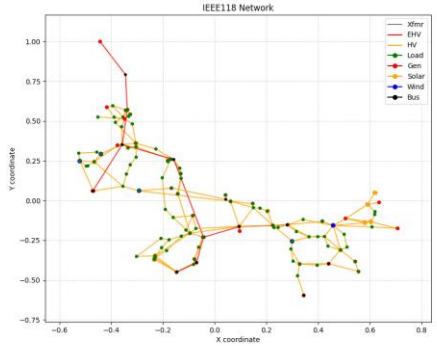
51-page body

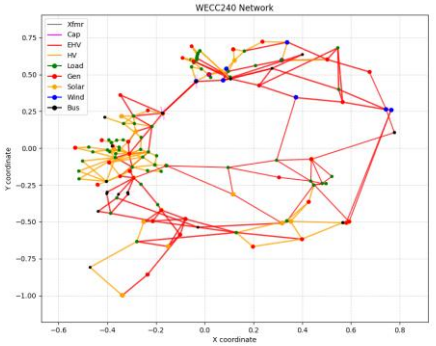
An extension and profile of the CIM “works” on five published examples of EMT cases; they all run *raw/dyr* ➔ *CIM* ➔ *ATP*.

Component Counts	Xfmr Sat	IEEE 39	IEEE 118	WECC 240	SMIB DLL
Buses	5	39	193	333	12
Machines	0	9	56	105	0
Solar	0	1	14	25	1
Wind	0	0	5	10	0
DLL	0	0	0	0	1
Notes	Load rejection with non-linear transformer	New England circa 1979	AEP System circa 1962	w/ Series Capacitors 2 aggregate DER By NLR, circa 2020	SMIB test harness for DLL

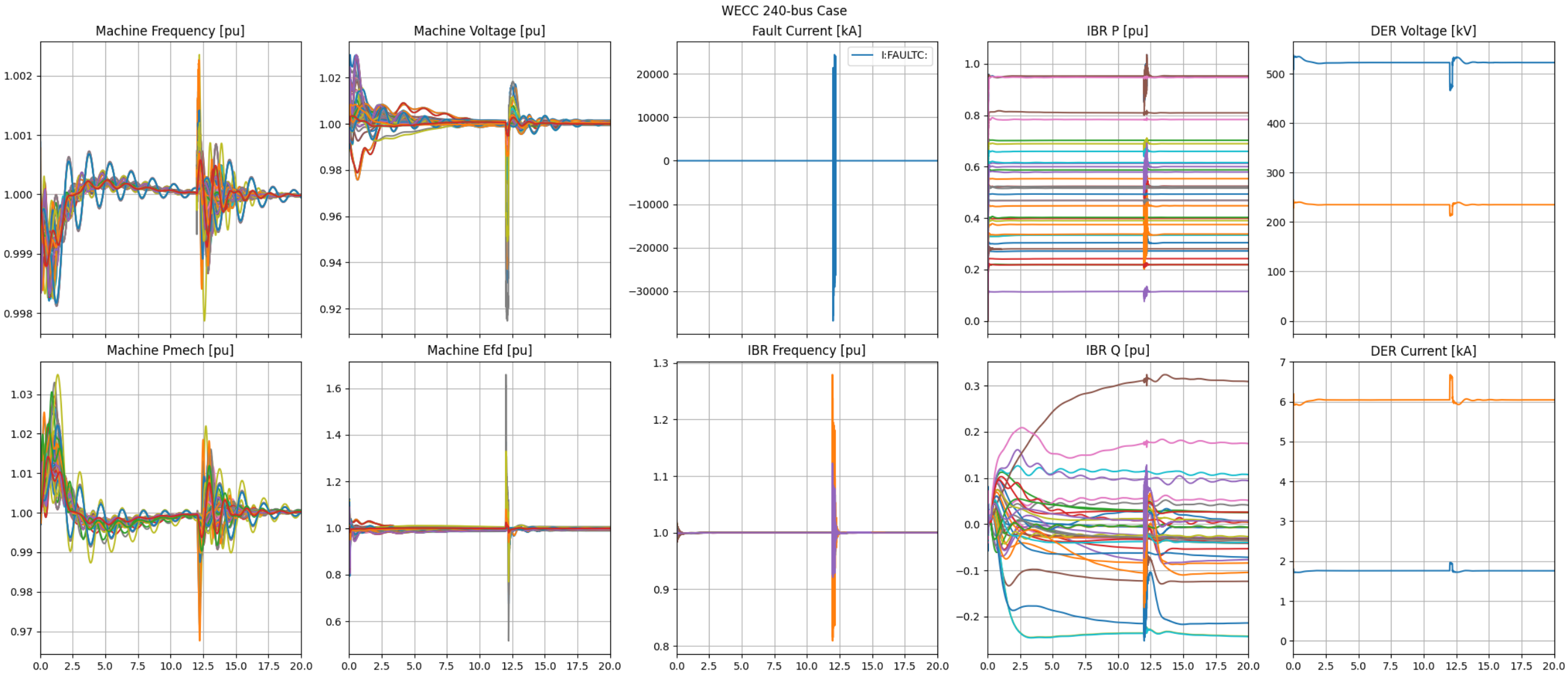
* Some extra buses for generator step-up (GSU) transformers.



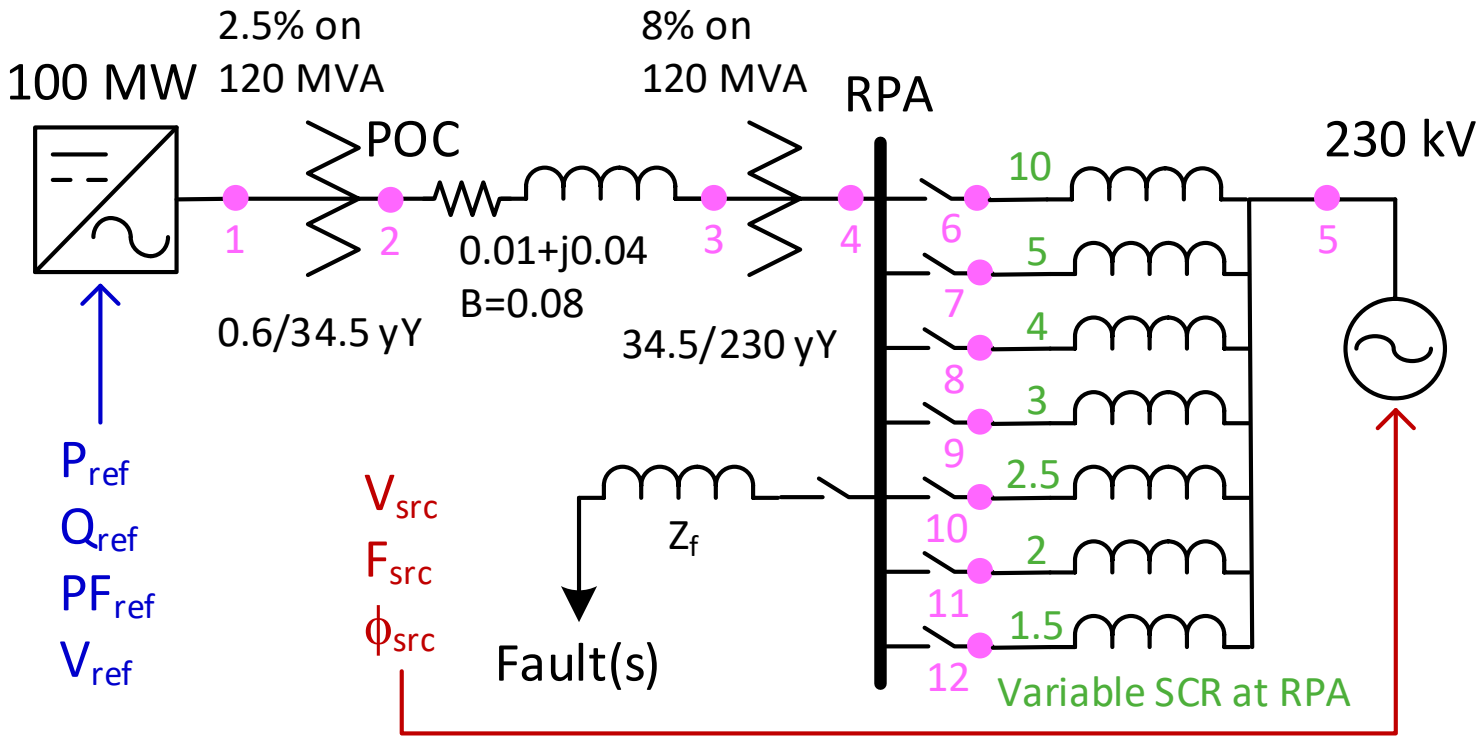


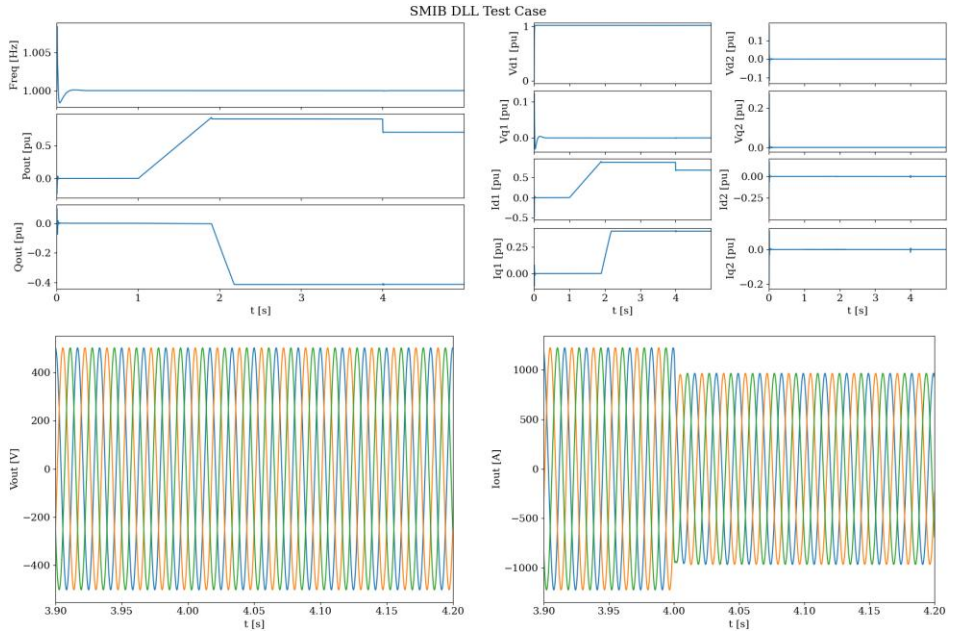


Sample ATP results for the WECC 240-bus system include 140 dynamic responses to a SLGF with 21% IBR generation. Initialized by MATPOWER.



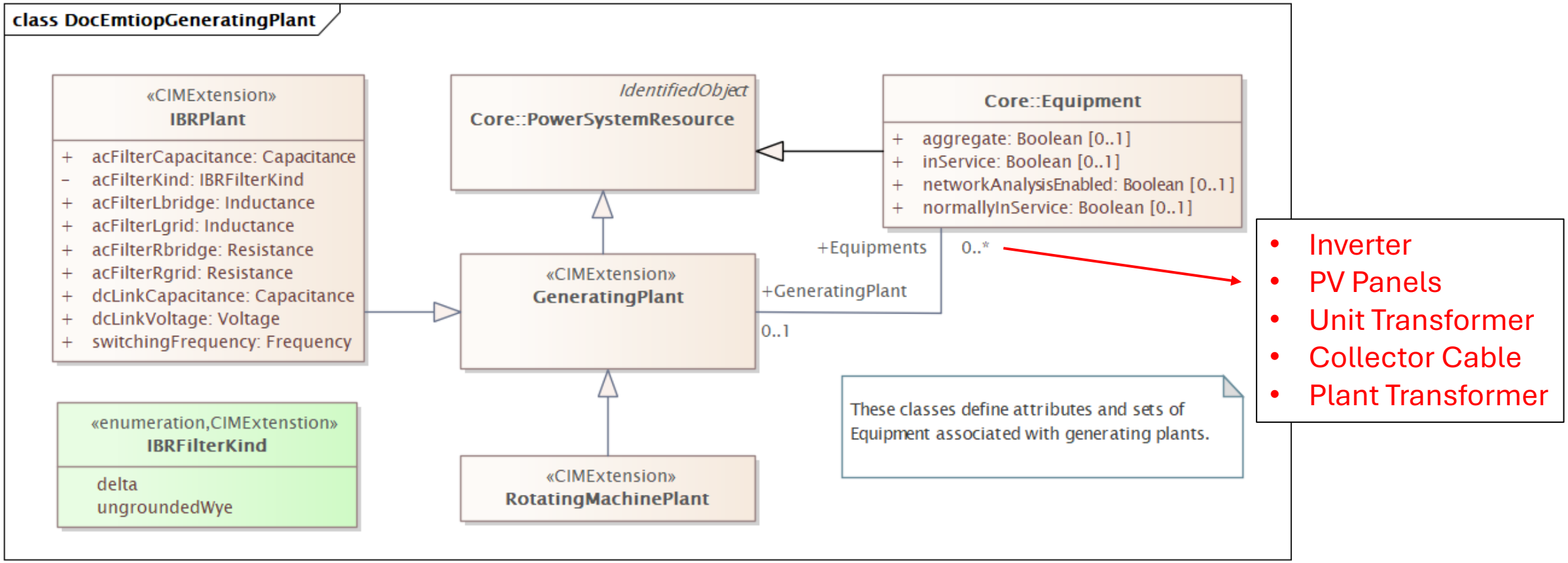
The CIM extensions for the DLL interface have been tested in a small-machine, infinite bus (SMIB) with an EPRI DLL (modified).





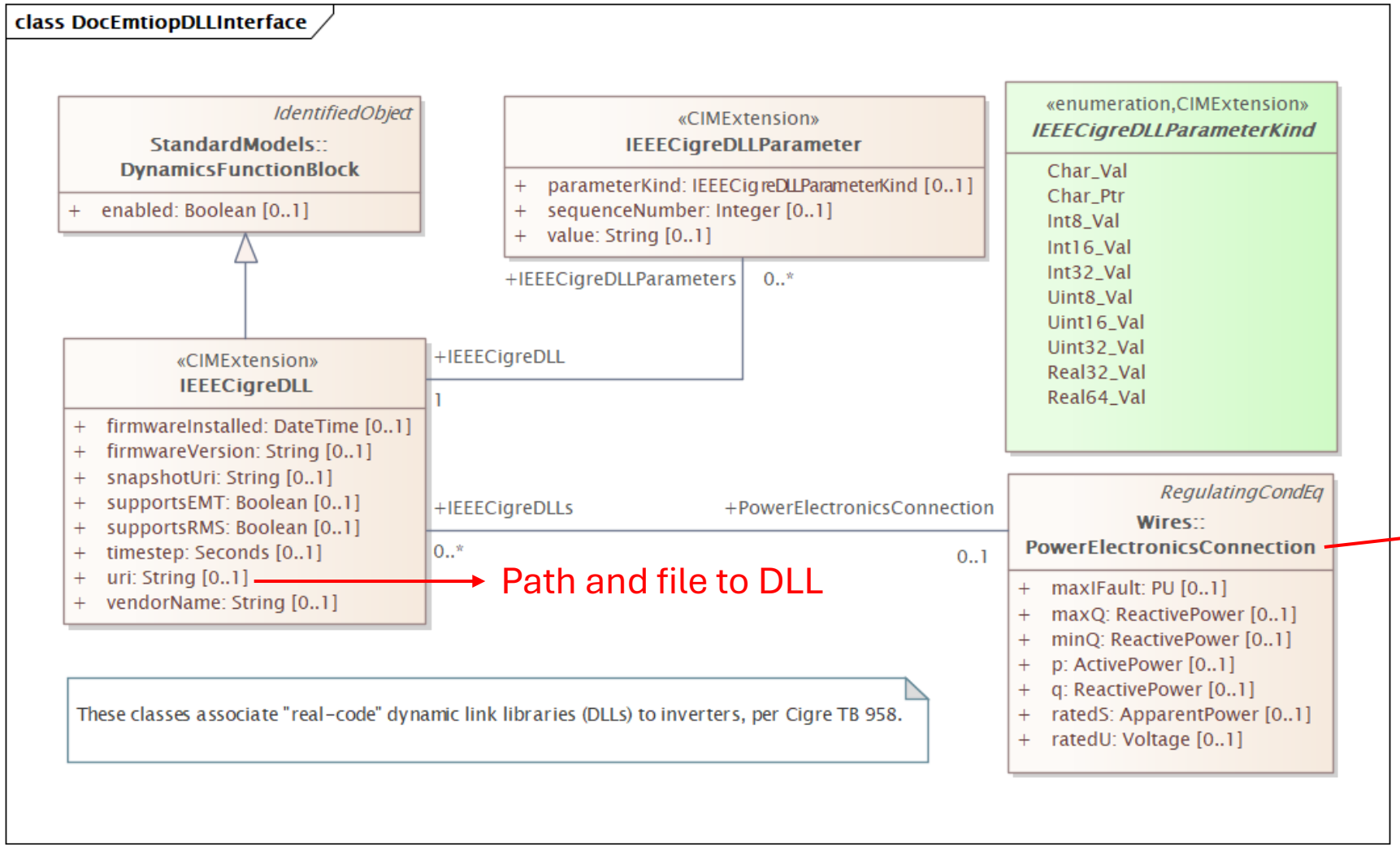
(Z_f and faulting switch are not in the example network model.)

For now, the new *IBRPlant* class aggregates components behind the RPA, with defined AC filter and DC bus attributes.

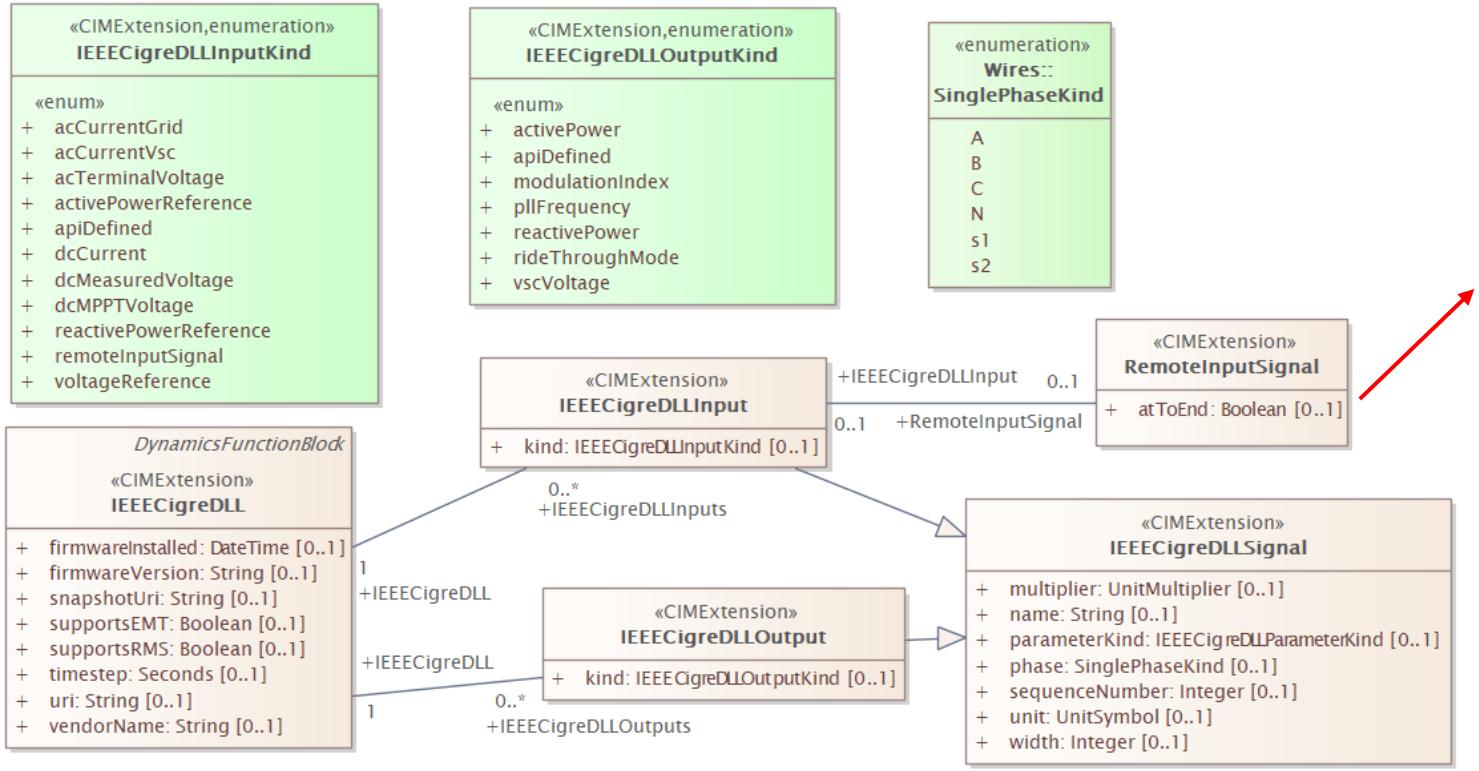


Also see NERC EMTWG WI9, “Draft White Paper: Uniform User Interface and Internal Structure for IBR Plant Models,” April 2016.

The new *IEEECigreDLL* class has an associated sequenced array of *IEEECigreDLLParameters* and *PowerElectronicsConnection*.

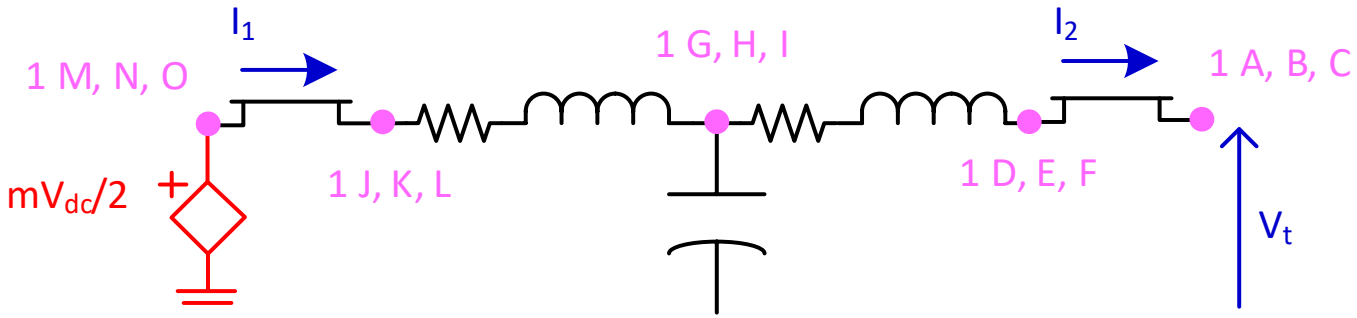


For now, *IEEECigreDLLInput* and *IEEECigreDLLOutput* signals assume the same internal nodes as EPRI's example DLL.



This existing CIM class is an option, if we model the individual filter components in the CIM, then include those in the new *IBRPlant* class aggregation of components.

(DC bus is optional, not included in this example.)



This is what CIM Resource Description Framework (RDF) data looks like, in the Turtle (TTL) format.

An *IBRPlant*

```
<298095DE-85A1-474C-8316-8CAE0F932952> a emt:IBRPlant ;
  emt:GeneratingPlant.Equipments <0A82F415-8F69-431F-8CA1-9561F2857828>,
    <3B6C281D-69C1-4E53-8AD3-60B5D42BA2A5>,
    <553C8DD4-48BB-4255-994B-6951693FEC63>,
    <B720A0C7-D1AB-43FE-86E0-190BAB17F6DC> ;
  emt:IBRPlant.acFilterCapacitance "0.0015"^^cim:Capacitance ;
  emt:IBRPlant.acFilterKind emt:IBRFilterKind.ungroundedWye ;
  emt:IBRPlant.acFilterLbridge "0.0001"^^cim:Inductance ;
  emt:IBRPlant.acFilterLgrid "0.0"^^cim:Inductance ;
  emt:IBRPlant.acFilterRbridge "0.00075"^^cim:Resistance ;
  emt:IBRPlant.acFilterRgrid "0.0"^^cim:Resistance ;
  emt:IBRPlant.dcLinkCapacitance "0.1"^^cim:Capacitance ;
  emt:IBRPlant.dcLinkVoltage "1200.0"^^cim:Voltage ;
  emt:IBRPlant.switchingFrequency "3060.0"^^cim:Frequency ;
  cim:IdentifiedObject.mRID "298095DE-85A1-474C-8316-8CAE0F932952"^^cim:String ;
  cim:IdentifiedObject.name "1_1"^^cim:String .
```

A DLL and its first parameter (happens to be the base V_{LL})

```
<FDE751E2-C11C-485C-B254-52B43FE55028> a emt:IEEECigreDLL ;
  emt:IEEECigreDLL.PowerElectronicsConnection <0A82F415-8F69-431F-8CA1-9561F2857828> ;
  emt:IEEECigreDLL.firmwareInstalled "2026-02-27T00:00:00"^^cim:DateTime ;
  emt:IEEECigreDLL.firmwareVersion "1.1.0.5"^^cim:String ;
  emt:IEEECigreDLL.snapshotUri ""^^cim:String ;
  emt:IEEECigreDLL.supportsEMT "true"^^cim:Boolean ;
  emt:IEEECigreDLL.supportsRMS "false"^^cim:Boolean ;
  emt:IEEECigreDLL.timestep "1e-05"^^cim:Seconds ;
  emt:IEEECigreDLL.uri "C:\\src\\emthub\\dll\\bin\\gfm_gfl_ibr2.dll"^^cim:String ;
  emt:IEEECigreDLL.vendorName "EPRI, IEEE EMTIOP WG"^^cim:String ;
  cim:DynamicsFunctionBlock.enabled "true"^^cim:Boolean ;
  cim:IdentifiedObject.mRID "FDE751E2-C11C-485C-B254-52B43FE55028"^^cim:String ;
  cim:IdentifiedObject.name "1_1"^^cim:String .

<FDE751E2-C11C-485C-B254-52B43FE55028_1> a emt:IEEECigreDLLParameter ;
  emt:IEEECigreDLLParameter.IEEECigreDLL <FDE751E2-C11C-485C-B254-52B43FE55028> ;
  emt:IEEECigreDLLParameter.parameterKind emt:IEEECigreDLLParameterKind.Real64_Val ;
  emt:IEEECigreDLLParameter.sequenceNumber "1"^^cim:Integer ;
  emt:IEEECigreDLLParameter.value "600.0"^^cim:String .
```

A handful of open questions before P3743 could move to ballot.

1. Switch to a full-CIM approach, including *Terminals*? **Voting May 14**.
2. Rely fully on CIM's *RemoteInputSignal* for the DLLs? **Voting May 14**.
   a) This means modeling individual DC and AC filter components in CIM
   b) We need IEEE P3597 to standardize the DLL input & output names
3. What essential North American excitation system, power system stabilizer, governor, and other controllers must be in CIM? **For June 11**.
4. Construct a smaller CIM profile with just IBR plant components and DLL, to connect with an existing non-CIM network model. **For June 11**.
5. Some nice-to-have items that should not delay balloting:
   a) EMT outputs: recommend CSV, COMTRADE, COMNAME, and/or CIM?
   b) Informative Annex on Large-Load EMT modeling (suggest CIM + DLLs)
   c) Informative Annex on mutual line coupling in zero sequence (*.seq files)