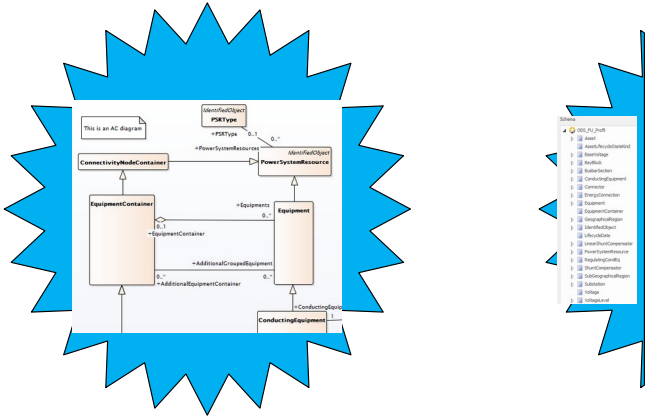
IEC Common Information Model

KONČAR
Inspired by challenge

Hedge-IoT 2025 April

IEC Common Information Model



Model Profile Serialization

KONČAR

Common Information Model

- Common information model for electrical network data exchange
  - Set of open standards for representing power system components
- Used mostly for:
  - **network model and case data exchange** (IEC61970)
  - electrical energy markets (IEC62325)
  - business process integrations and messaging (IEC61968)
- Only defines the semantics and the data exchange
  - storage and model data manipulation is not in scope
- Developed by IEC TC57 WG 13/14/16
  - Using Enterprise Architect tool
  - https://cimug.ucaiug.org

Common Information Model

- CIM uses an **UML** based ontology
  - Classes, attributes and relations with multiplicity
  - Single class inheritance
  - Use of 128-bit UUID unique identifiers (MRID)
  - Extendable by users
- Data is encoded in **RDF** data structure
  - Class instance – attribute – value triplets
- Syntax and semantic checks and validation
  - XSD schemas, RDFS, OWL/OCL, SHACL
  - application validation

Terminology

- SCADA world
  - Measurements (time series samples) = (measurement) data
  - CIM model data (RDF object data) = (network) model
  - CIM information model (UML/RDFS definitions) = (network) metamodel
  - CIM header data = (network) model metadata
- Knowledge graph world
  - RDF object data = data
  - UML/RDFS definitions = model

CIM model characteristics

- (Not only) object oriented
  - Can be interpreted as a relational, document or graph model
- Every instance has a class and unique id (mRID UUID)
- Object instances correspond to:
  - physical elements (lines, transformers, switches...)
  - organizational elements (station, region, voltage level)
  - topological elements (connectivity node, terminal)
  - types (transformer, line, switch...)
  - measurement locations, GIS shapes
  - etc.
- Class hierarchy and attribute semantics prescribed by the basic standard
- CIM can be extended
  - custom classes, attributes and profiles

KONČAR

CIM serialization

- Standardized rendering of RDF model data into files/messages
  - CIM/XML (IEC61970-552)
  - Turtle ("master format", but rarely used in practical exchanges)
  - JSON-LD (future IEC61970-553 standard)
- mRID (UUID) attributes represent relations with other objects
- CIM/XML is very similar to RDF/XML
  - Unfortunately, not completely compatible with RDF/XML
  - W3C recommendations were not complete at the time
  - Control center standards, software and processes are very slow to upgrade
- Headers encoded separately
  - switching to (extended) DCAT in the future
- CIM has its own data types
  - mapped to corresponding XML/XSD types

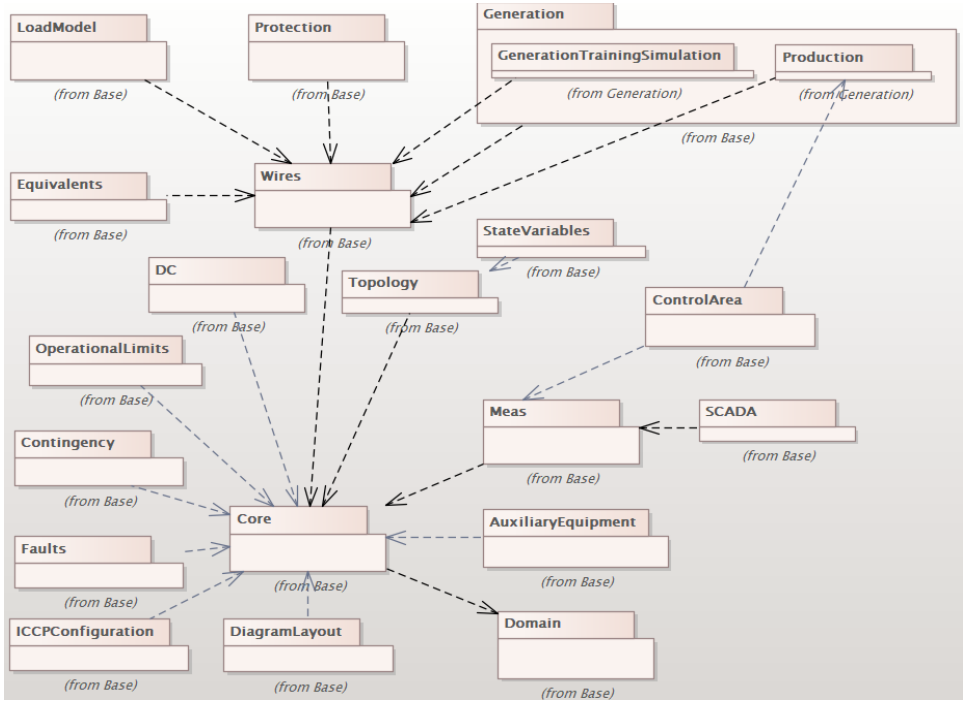
KONČAR

CIM profiles

- Profile = subset of complete CIM model for a specific use-case
  - List of included concrete classes
  - List of mandatory and optional attributes/relations per class
- RDF schema (**RDFS**) for IEC 61970 standard group
- XML schema (**XSD**) for IEC 61968 and 62325 standard groups
- Additional constraints (natural language, OWL, OCL, **SHACL**)
- Profiles for equipment (EQ), topology (TP), load flow (SSH, SV), geographical layout (GL), diagram layout (DL), dynamics (DY)...

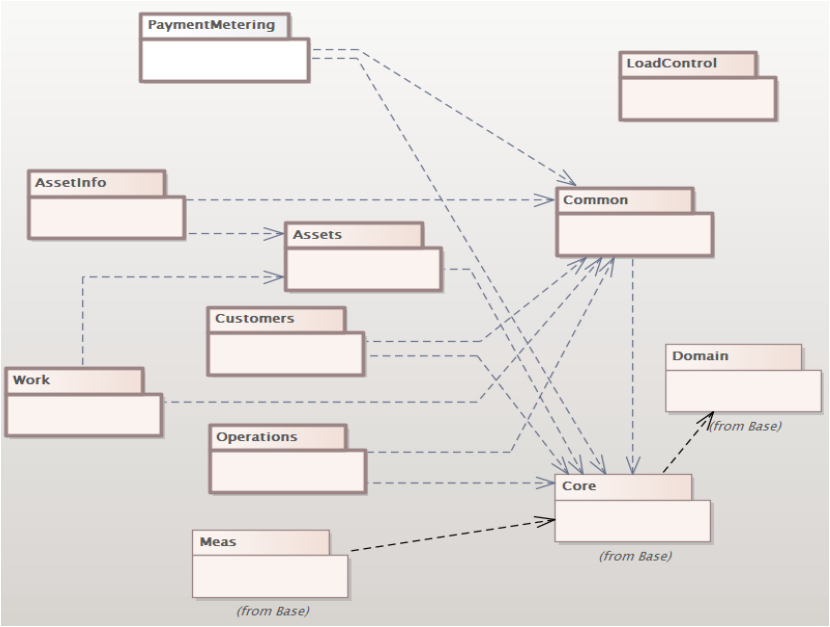
IEC TC57 WG13 – IEC61970 (TSO/network modelling/grid)

- Core CIM part
- Network model for basic EMS network calculations (load flow, topology, short circuit...)
- Dynamic – element models for transient response modelling
- Meta – data exchange metadata, CRUD change modelling



IEC TC57 WG14 – IEC61968 (DSO/business integration/support)

- 61968-3 Interface for network operations
- 61968-4 Interfaces for asset management
- 61968-5 Distributed energy optimization
- 61968-6 Interfaces for maintenance and construction
- 61968-8 Customer Operations
- 61968-9 Meter reading and control



Why CIM ?

- CIM models most elements and processes within electric energy sector
- CIM includes many engineer-years of built-in domain expertise
- Any integration of systems creates an implicit common information model
- CIM is extensible by users
- CIM enables interoperability with industry standard software
  - SCADA systems
  - Standalone network modelling software (PSS-E, NEPLAN, Digsilent...)
  - GIS systems, workforce management, market systems...
  - Asset management systems
  - TSO mandatory integrations with ENTSO-e software
- Enables modularity, phasing of changes and incremental system replacement

CIM modelling and concepts

- IdentifiedObject
  - base class for all named objects
  - mRID UUID and multiple system specific names (Name/NameType/NameTypeAuthority)
- Class structure and inheritance
  - Breaker ➔ Switch ➔ ConductingEquipment ➔ Equipment ➔ PowerSystemResource ➔ IdentifiedObject
- Container classes
  - Regions, substations, voltage levels

CIM PSR

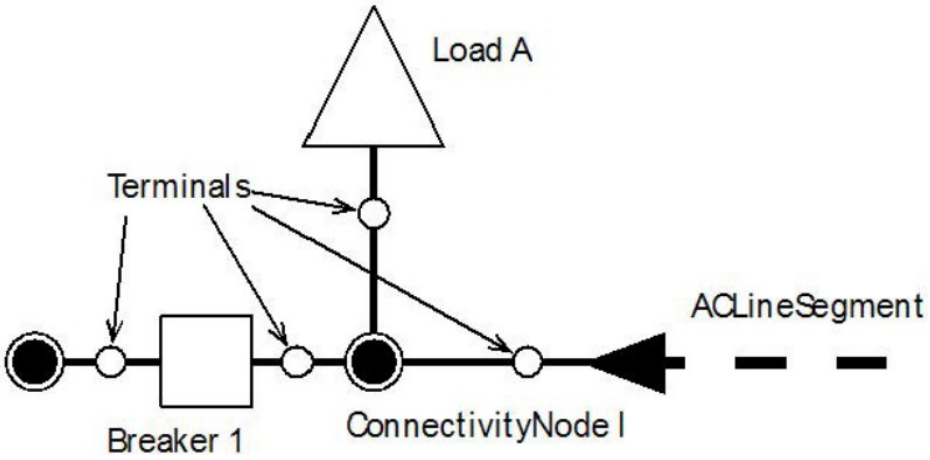
- PSR-Asset-AssetInfo
  - Enables correct modelling of assets separate from functions and characteristics
- PowerSystemResource
  - functional role of element in network
  - "transformer in abc substation connected to..."
- Asset
  - specific physical unit with serial/inventory number etc...
  - can be connected in network, stored
- AssetInfo
  - "nameplate" for a type of unit with common characteristics

CIM modelling and concepts

- Document headers
  - Metadata for exchanged files/messages
  - Source, creation time, scenario time, description, SupersededBy...
- Diagrams
  - DiagramLayout profile
  - basic (SLD) diagram exchange
  - vendor neutral, only defines the locations of symbols and lines
  - styling is provided by the presenting application
- Geographic data
  - Location, PositionPoint and CoordinateSystem classes
  - coordinate system + array of point coordinates
  - points, linestrings, polygons
  - compatible with basic GIS (WKB) shape objects

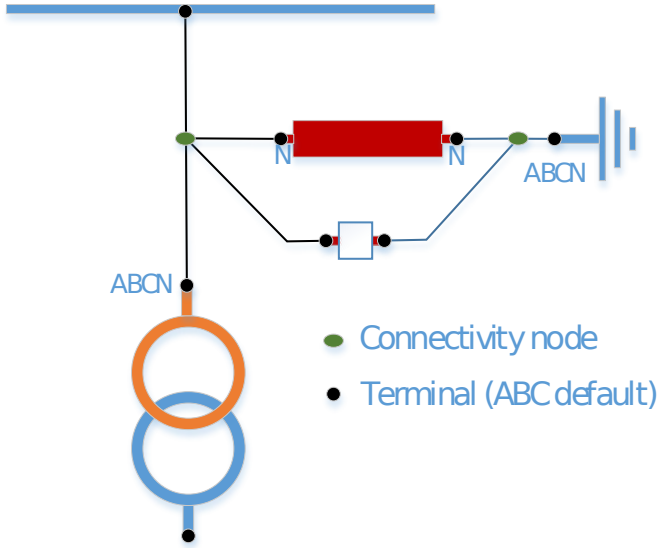
CIM modelling and concepts

- Terminal and ConnectivityNode classes define connections



CIM modelling and concepts

- Connectivity nodes and terminals have phase information



CIM difference model

- incremental data exchange
  - incremental exchange of CRUD model changes
  - SupersededBy header field holds the base model reference
- forward and reverse statements
  - used instead of explicit CR-U-D annotations
  - not true forward/reverse diff, both are always needed
  - forward - object creation and modification
  - reverse - object deletion and (reverse) modification

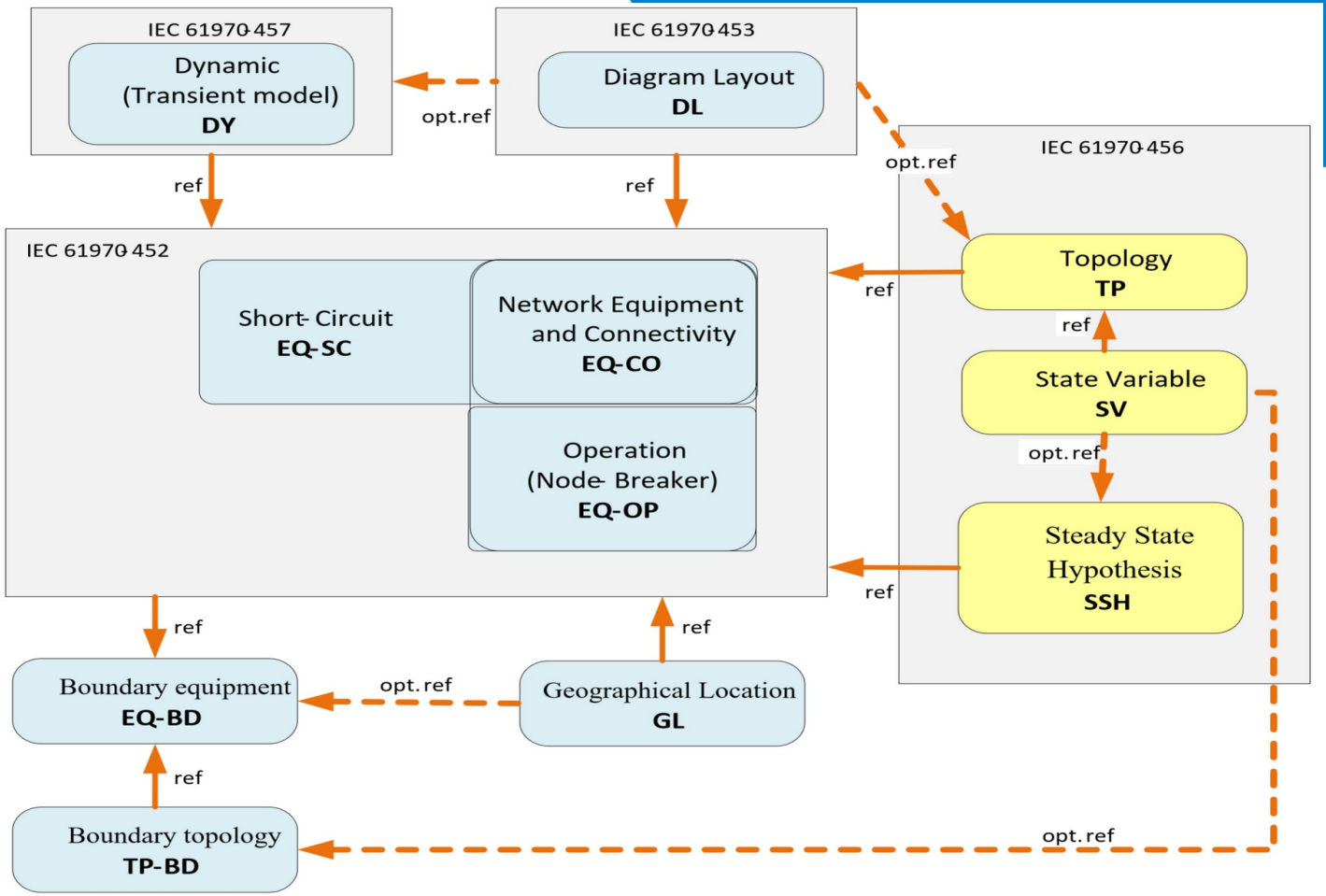
CGMES

- Common Grid Model Exchange Standard
  - Concrete CIM profile for ENTSO-E use cases
  - CGMES v2.4 = CIM v16 UML + profiles, v3.0 still not implemented
  - IEC 61970-452 (EQ), -456 (TP, SV), -453 (DL), -457 (DY), IEC61968-13 (GL)
  - Mandatory interface for data exchange between ENTSO-E TSOs
  - Enables data exchange and merging of individual TSO network models
  - Data exchange between TSOs and individual applications within same TSO

CGMES

- Applications dealing with:
  - power system data management
  - load flow
  - contingency analysis
  - short circuit calculations
  - market transparency
  - capacity allocation and congestion management
  - planning and network upgrades
  - dynamic security assessment

CGMES


IEC 61970-457
Dynamic
(Transient model)
DY
opt.ref
IEC 61970-453
Diagram Layout
DL
opt.ref
IEC 61970-456
ref
ref
IEC 61970-452
Short- Circuit
EQ-SC
Network Equipment
and Connectivity
EQ-CO
Operation
(Node- Breaker)
EQ-OP
ref
Topology
TP
ref
State Variable
SV
opt.ref
Steady State
Hypothesis
SSH
ref
ref
ref
Boundary equipment
EQ-BD
opt.ref
Geographical Location
GL
ref
Boundary topology
TP-BD
opt.ref

CIM developer perspective

- CIM is unavoidable in the electric power sector today
- Core idea (common information model) is very good
- CIM doesn't solve the problem of application data structures, data manipulation and database storage
- Lots of minor problems:
  - design-by-committee, slow adoption of standards
  - complex standard, software not fully interoperable in practice
  - people involved in standards are mostly not professional software engineers
  - using many cool semantic web standards (RDF, JSON-LD)
  - while making incompatible modifications and a bad/convoluted implementation on top
  - which needs to be revised a few years later...

CIM developer perspective

- Lots of minor problems:
  - breaking changes (CGMES 2/3, headers, naming, OCL/SHACL...)
  - problems with complex datatypes and measurement units
  - problems with prefixes, namespaces
  - problems with mRID (UUID or not?), serialization prefixes
  - problems with difference model (deletes without old value)
  - good and performant CIM software tooling is missing (esp. opensource)
  - standalone object granularity
  - various modeling errors