

Compositional models for power systems

John S. Nolan¹, Blake S. Pollard², Spencer Breiner², Dhananjay Anand², and Eswaran Subrahmanian³

¹University of Maryland, College Park, MD, USA

²National Institute of Standards and Technology, Gaithersburg, MD, USA

³Carnegie Mellon University, Pittsburgh, PA, USA

The problem of integrating multiple overlapping models and data is pervasive in engineering, though often implicit. We consider this issue of model management in the context of the electrical power grid as it transitions towards a modern ‘Smart Grid.’ We present a methodology for specifying, managing, and reasoning within multiple models of distributed energy resources (DERs), entities which produce, consume, or store power, using categorical databases and symmetric monoidal categories. Considering the problem of distributing power on the grid in the presence of DERs, we show how to connect a generic problem specification with implementation-specific numerical solvers using the paradigm of categorical databases.

1 Introduction

The modeling of complex systems, engineered or natural, entails certain generic challenges: the existence and interaction of multiple models, multiple algorithms, and multiple implementations. This paper presents a methodology rooted in category theory to manage this complexity, concretized via a model-driven engineering approach to designing a modern electrical grid, dubbed the ‘Smart Grid.’

The existing grid architecture is characterized by dedicated large-scale, centralized generation and distributed, downstream consumption. Moving towards an architecture with increased distributed generation will have a profound impact on how the grid is managed: end users will no longer be dedicated

consumers, but will shift between consuming and producing power. One key to enabling this transition is the management and modeling of distributed energy resources (DERs), generic devices that can consume, produce, or store power.

The notion of DER is meant to provide an abstraction or characterization summarizing the essential properties of a wide array of different energy resources, e.g. photovoltaic systems, batteries, conventional loads, and so on. The issue is that no uniform abstraction exists. Different stakeholders utilize different abstractions for different purposes. In addition, these meta-models must evolve as new technologies emerge.

Coupled with control mechanisms, DERs provide a number of ancillary services to consumers and grid operators: voltage control, reducing peak loads, demand response, etc. [15, 18]. Aggregations of heterogeneous DERs provide the abstraction through which such collections participate in the overall power system and energy markets [4].

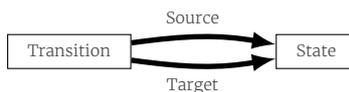
There is a large body of work concerned with the use of model transformations in the context of model management and model-driven engineering, e.g. [13, 24]. A number of approaches utilize category theory, recognizing the natural mathematical framework it provides for reasoning about models, their semantics, and structure-preserving transformations among models [5, 25].

We tackle the problem of specifying, relating, and transforming models using the functorial data model advocated in [3, 8, 19, 20, 22] as well as its computational implementation in the CQL tool. In the functorial data model, database schemas are interpreted as finite presentations of categories. Instances of a database schema correspond to Set-valued functors out of the associated category. Some subtleties arise when working with computational data such as strings and integers, though we will not concern ourselves with these difficulties; see [20] for a thorough treatment.

Structure of this paper: In Section 2, we consider the problem of model specification and transformation for family of DER schemas and the functors among them. In Section 3, we show that one variant of a model in this family yields the objects of a symmetric monoidal category with DER aggregation as monoidal product. In Section 4, we consider the problem of distributing power in a grid where certain nodes correspond to aggregate collections of DERs, describing a procedure for translating among numerical solvers using database schemas, functors, and queries.

2 Categorical databases for model management

The family of models we present in this section share a common ancestor, the directed multigraph, henceforth graph, consisting of two entities, States and Transitions, together with two arrows, Source and Target, assigning source and target states to transitions:



Including identities and composites, this schema forms a category DiGraph . Functors from this category to Set form a category of instances.

2.1 A basic DER model

In our base model, DERs are viewed as graphs with operational states as nodes and transitions among those. Each state is assigned a feasible operating region (i.e. power demands / generation). In AC circuits, power is a complex-valued quantity $P + iQ$, where the real part P is referred to as real or active power and the imaginary part Q as reactive power. For now we restrict our attention to the case where operating regions are single points.

Our base DER model is described by the schema DER_{Base}



consisting of a graph together with two attributes for each state, $P, Q: \text{State} \rightarrow \text{Float}$. Two instances of DER_{Base} are depicted in Figure 1, showing a typical load (an HVAC, i.e. heating/cooling system) and a battery.

2.2 Model Translation via Functors

Depending on the analysis to be performed, this basic DER meta-model or schema extends to include additional information such as state of charge, virtual cost of transitions, location, etc. The functorial data model offers a robust collection of ways to translate between such models. Some of these models and the functors relating them are summarized in the following and in Figure

2.

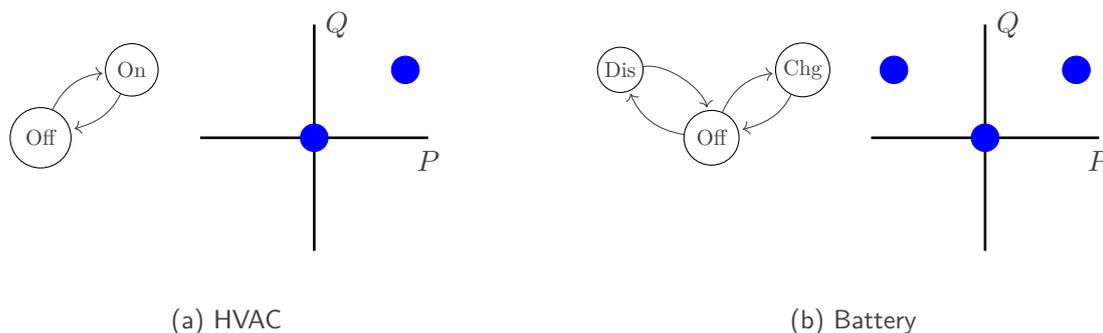


Figure 1: Two types of DERs and their associated demand profiles. With the chosen convention, states with positive real power P consume power, while states with negative P generate power.

In [1, 16], virtual costs are assigned to transitions representing the willingness/ability of a controllable DER to perform a certain transition. This leads to a new schema DER_{Cost} with an additional attribute for transitions and obvious inclusion functor to it from the base DER model.

Attaching non-negative rates to transitions on a graph gives a Markov process, summarized by the schema Markov [14]. For now, we implement these rates as ‘Floats,’ tabling the discussion of constraints in CQL until Section 4.2. Quantifying variability of generation from renewable sources is a key issue when modeling DERs. One approach models DERs as Markov processes, giving a stochastic DER model DER_{Mark} .

Over long time scales, the steady state probabilities of such a model can be used to estimate energy production and other performance indices. In [12], this approach is utilized to evaluate reliability of small wind farm generation by assigning probabilistic transitions between operative and failed states and coupling this with a stochastic model of wind variability. This methodology is also applied to small hydro electric stations in [2]. Stochastic models of solar irradiance are also used to generate synthetic data for system design [26]. We summarize these stochastic models of weather in the schema Weather .

Functors $F: M \rightarrow N$ between database schemas give rise to adjoint triples of functors $\Sigma_F, \Delta_F, \Pi_F$ between the associated categories of instances, where $\Delta_F: N\text{-Inst} \rightarrow M\text{-Inst}$ and $\Sigma_F, \Pi_F: M\text{-Inst} \rightarrow N\text{-Inst}$. These functors are related to “uber-flower” queries $Q: M \rightarrow N$. Such a Q can be evaluated (as in other data models) to give a functor $\text{eval}(Q): M\text{-Inst} \rightarrow N\text{-Inst}$ or dually “coevaluated” to give a functor $\text{coeval}(Q): N\text{-Inst} \rightarrow M\text{-Inst}$. Using data migration functors and queries, along with CQL’s ability to compute colimits of instances, offers

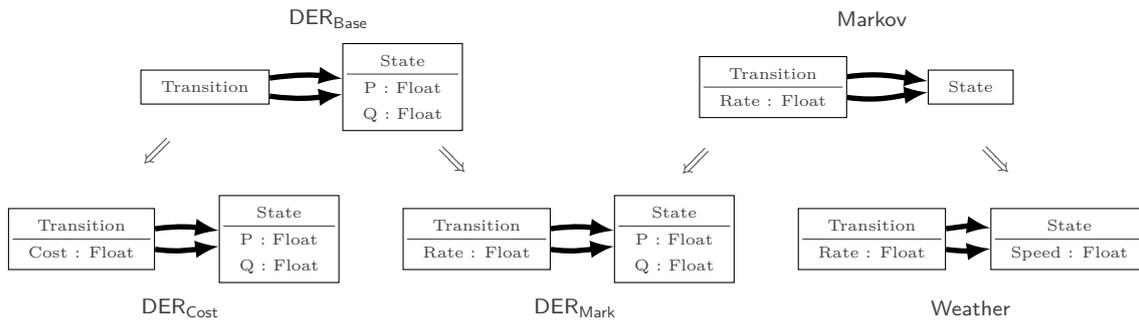


Figure 2: Multiple model schemas (boxes are objects, dark arrows are morphisms), connected by functors (the hollow arrows). On the left, DERs with costly transitions share a common underlying base model with a stochastic DER model. On the right, stochastic DER models and stochastic models of natural processes are both modeled using Markov processes.

a useful way to translate between the data associated to different models.

The models presented here are connected by inclusion functors, summarized in Figure 2. Even in the simple setting described here, the use of mappings connecting DER models enables reuse and validation of said models, while providing an extensible framework for model documentation. Implementing mappings between models which utilize the same method, such as Markov processes, enables the reuse of tools or methods, e.g. steady state solvers. In Section 4.4, we give a more detailed exposition on how functors between schemas along with their associated adjoint triples and queries can be used to connect models to tools within the paradigm of categorical databases.

In the next section we give a categorical treatment of DER aggregation, the process of taking collections of DERs and combining them into a single DER. For this we move to the setting of symmetric monoidal categories and consider demand regions as subsets of the complex plane.

3 Aggregation as a Symmetric Monoidal Product

Aggregation of DERs is the key to unlocking their potential to provide ancillary services such as peak shaving and voltage control. It is often third-party aggregators who act as intermediaries between utilities and customers, pooling resources and providing data integration and control strategies via the development of distributed energy resource management systems (DERMS) [7].

In this section, we focus on the basic problem of aggregating demand re-

gions and state spaces of DERs, presenting a symmetric monoidal category DER whose objects are DERs, and where aggregation serves as the tensor product. This model adds a reflexive property [23], i.e. mandatory self-edges for each node, to the underlying graphs of the DER models outlined previously. Morphisms in DER correspond to adjusting the level of granularity of the state space.

Definition 1. A *distributed energy resource (DER)* $\mathcal{D} = (S, T, s, t, r, d)$ consists of a graph $s, t: T \rightarrow S$, together with a function $r: S \rightarrow T$, satisfying $s \circ r = t \circ r = \text{id}_S$, picking out an *identity transition* from each state to itself, and a function $d: S \rightarrow 2^{\mathbb{C}}$ assigning to each state $\sigma \in S$ a *power demand region* $d(\sigma) \subseteq \mathbb{C}$. For each state $\sigma \in S$ we write $1_\sigma = r(\sigma)$ and call 1_σ the identity transition of σ .

This definition is summarized by the diagram $T \begin{array}{c} \xrightarrow{s} \\ \xleftarrow{r} \end{array} S \xrightarrow{d} 2^{\mathbb{C}}$.

Definition 2. A *morphism of DERs* $\phi: \mathcal{D} \rightarrow \mathcal{D}'$ consists of a pair of functions (ϕ_S, ϕ_T) , where $\phi_S: S \rightarrow S'$ and $\phi_T: T \rightarrow T'$, such that for all $\tau \in T$, $\phi_S(s(\tau)) = s'(\phi_T(\tau))$ and $\phi_S(t(\tau)) = t'(\phi_T(\tau))$, and for all $\sigma \in S$, $\phi(1_\sigma) = 1_{\phi(\sigma)}$ and $d(\sigma) \subseteq d'(\phi_S(\sigma))$. Together with these morphisms (and the obvious identity morphisms and composition law), DERs form a category which we denote DER.

In short, a morphism of DERs is a homomorphism of the underlying graphs that acts as an inclusion of subsets on the demand regions for each state. Such morphisms can be used to translate between models of a DER, e.g. by adding more states or by merging states which are indistinguishable in the codomain model. An example of this is provided in Subsection 3.1.

Demands can be aggregated using Minkowski sums; see [6] for more details as well as [10] for an application to modeling the flexibility of DERs.

Definition 3. Given two subsets $X, Y \subseteq \mathbb{C}$, the *Minkowski sum* of X and Y is the set

$$X + Y = \{x + y : (x, y) \in X \times Y\} \subseteq \mathbb{C}.$$

Under this operation, $2^{\mathbb{C}}$ is a commutative monoid with unit $\{0\}$.

Definition 4. The *aggregate* of two DERs \mathcal{D} and \mathcal{D}' is the DER $\mathcal{D} \otimes \mathcal{D}' = (S \times S', T \times T', s \times s', t \times t', r \times r', d + d')$, where $d + d': S \times S' \rightarrow 2^{\mathbb{C}}$ is defined by $(d + d')(\sigma, \sigma') = d(\sigma) + d'(\sigma') \subseteq \mathbb{C}$ for any $(\sigma, \sigma') \in S \times S'$.

In short, the aggregate of two DERs is the categorical product of the underlying graphs (see [23] or [17, Proposition 3.3.9]), where each product state



is equipped with demand equal to the Minkowski sum of its factors. Observe that the reflexive property of individual DERs within an aggregate DER enables independent transitions.

Aggregation extends easily to morphisms by taking Cartesian products of functions, so in this way we see $\otimes: \text{DER} \times \text{DER} \rightarrow \text{DER}$ is a bifunctor. In fact, letting \mathcal{I} denote the DER with one state σ , a single transition 1_σ , and power demand $d(\sigma) = \{0\} \subseteq \mathbb{C}$, it is not hard to show that DER is a symmetric monoidal category with tensor product \otimes and unit \mathcal{I} . As a result, string diagrams can be used to reason about DERs and aggregation [9, 21].

3.1 Net Demand Quotient

When aggregating DERs, the state space grows rapidly. For operations at the distribution level, all that is relevant is the *net* power demand. Thus it is natural to mod out by an equivalence relation whereby states with identical power demand are identified. The following definition formalizes this notion.

Definition 5. Let \mathcal{D} be a DER. Consider the equivalence relation \sim on the states S of \mathcal{D} where $\sigma \sim \sigma'$ if and only if $d(\sigma) = d(\sigma')$. This induces an equivalence relation \approx on the edges T of \mathcal{D} where $\tau \approx \tau'$ if and only if $s(\tau) \sim s(\tau')$ and $t(\tau) \sim t(\tau')$. We can define the **net demand DER** $\overline{\mathcal{D}}$ of \mathcal{D} by $\overline{\mathcal{D}} = (S/\sim, T/\approx, \overline{s}, \overline{t}, \overline{r}, \overline{d})$, where $\overline{s}, \overline{t}, \overline{r}$, and \overline{d} are defined in the obvious way.

The equivalence relation above gives rise to a DER morphism $\overline{(\)}: \mathcal{D} \rightarrow \overline{\mathcal{D}}$ which identifies states with equal power demand and transitions among them. Composing this morphism with aggregation applied to a pair of DERs \mathcal{D} and \mathcal{D}' gives a DER $\overline{\mathcal{D} \otimes \mathcal{D}'}$ which only distinguishes states which differ in their net power demand.

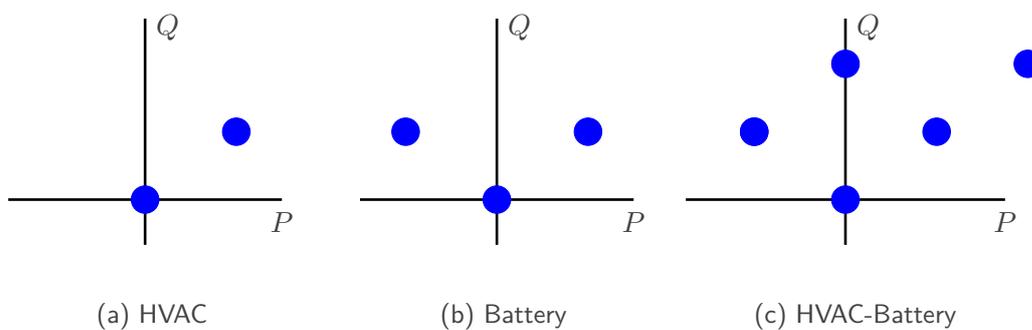


Figure 3: The demand profile hybrid or aggregate DER consisting of an HVAC system and a battery.

Any path in $\bar{\mathcal{D}}$ will give a set of paths in \mathcal{D} traveling among DER states. We can then consider methods for selecting the ‘best’ or ‘least-costly’ sequence of DER transitions which accomplish some desired transition in net demand. This allows for dynamic tasking of DERs to accommodate demand fluctuations without requiring distribution level operators to have full knowledge of the details of a collection of DERs.

DER aggregation is typically done locally/regionally, interfacing with grid operators at the distribution or transmission level where the problem becomes matching generation with consumption while maintaining stable operating conditions. We now turn to the basic problem of distributing electricity through the grid so as to match production and consumption. specification and numerical solution of basic power flow problems using categorical databases.

4 Power Flow Problems

In this section, we show how to connect models with tools or solvers by describing the specification and numerical solution of basic power flow problems using categorical databases. This amounts to finding solutions to a set of non-linear equations, the power flow equations, defined over a network or power flow graph:

Definition 6. A **power flow graph** consists of graph $s, t: E \rightarrow N$, together with functions $g, b: E \rightarrow \mathbb{R}$, assigning a **conductance** and **susceptance** to each edge. Nodes in the graph $n \in N$ are typically called **buses**, while edges $e \in E$ are referred to as **branches**. Conductance and susceptance are the real and imaginary parts of the complex admittance, a measure of the susceptibility of a branch to admitting current flow.

The variables of interest are the real and imaginary parts of the complex power $P + iQ$ and the magnitude and phase of the complex voltage $Ve^{i\theta}$, which we regard as partial functions $P, Q, V, \theta: N \rightarrow \mathbb{R}$. Buses are typed as PQ , PV , or $V\theta$ buses according to which pair of variables is regarded as fixed, see Figure 4. The remaining free variables are determined by solving the power balance equations.

Definition 7. The **power balance equations** [11] for a power flow graph are the $2|N|$ equa-

tions

$$P_i = V_i \sum_j V_j (g_{ij} \cos(\theta_i - \theta_j) + b_{ij} \sin(\theta_i - \theta_j))$$

$$Q_i = V_i \sum_j V_j (g_{ij} \sin(\theta_i - \theta_j) - b_{ij} \cos(\theta_i - \theta_j)),$$

where we write $P_i := P(N_i)$ and $g_{ij} := g(E_{ij})$ etc. and each sum is taken over all buses adjacent to i .

We summarize the data needed to specify a power flow problem in a CQL schema in Figure 4, omitting attributes for simplicity.

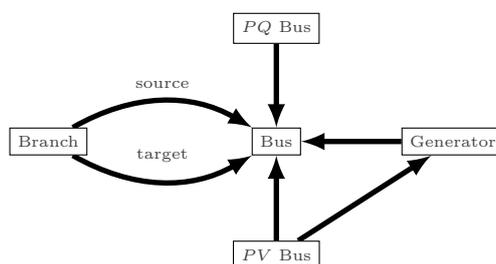


Figure 4: A schema describing a generic power flow problem. A PQ bus represents a typical load, whose real and reactive power demands are known and fixed, at any moment of time. All PV buses are viewed as having generators attached, producing constant power at a specific voltage. Slack buses are omitted for visual clarity.

Due in part to their non-linearity, solving the power flow equations is typically done numerically either using freely available software, commercial tools, or customized code. Such tools usually require specific solver parameters and use their own internal data structures.

4.1 Connecting to a Tool

MATPOWER is a commonly used power systems toolbox, implemented in MATLAB. The MATPOWER data format specifications are organized into tables in Appendix B of the MATPOWER manual [27]. We translate these specifications into MATPOWER-specific schemas in CQL. Figure 5 shows the resulting schemas representing both a power flow problem as well as an associated solver, e.g. We characterize an iterative Newton-Raphson solver in terms of its required parameters such as tolerance, maximum number of iterations, etc.

Encoding the input problem specification, the output solution structure, as well as the solver parameters in database schemas enables systematic experimentation, i.e. varying inputs or parameters, while providing flexible and

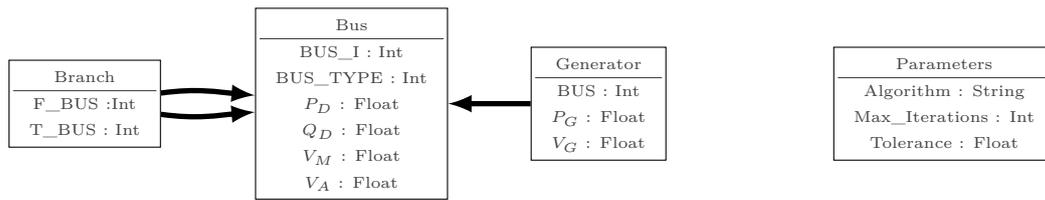
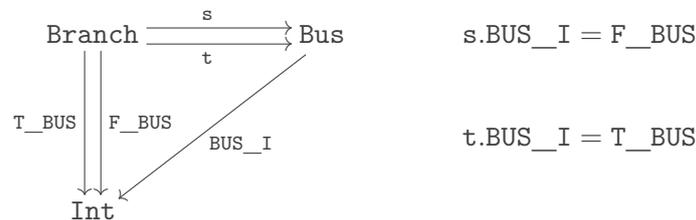


Figure 5: A MATPOWER power flow schema on the left, with solver parameters on the right. For simplicity we only show a few attributes for each entity. Attribute names are based on those in MATPOWER. Compare Figure 4, which was developed based on a reorganization of the schema on the left.

traceable documentation, i.e. storing just solutions or including the solver settings used in each run. The input and output features common to all solvers of a given type can be organized into a generic schema for solvers of that given type.

4.2 Constraints in CQL

CQL allows for the enforcement of constraints in the form of path equations. For example, consider the chunk of our MATPOWER schema:



The equations on the right enforce the constraint that the indexing of buses via BUS_I is consistent with the indexing of T_BUS and F_BUS of branches.

4.3 Connecting to DERs

To interface with a standard power flow problem, we place DERs at the relevant nodes of a power flow graph, treating each such node as a PQ bus. For each such bus we determine average P , Q values from the DERs at that node, for example by modeling the relevant DERs as Markov chains, as described in Subsection 2.1, and returning the sum of the expected steady-state P , Q values for each DER. This process is depicted in Figure 6 and implemented by the authors in a MATPOWER example.

This hybrid setup enables the exploration on the dependence of overall solutions to the power flow equations on the types and behaviors of DERs, e.g.

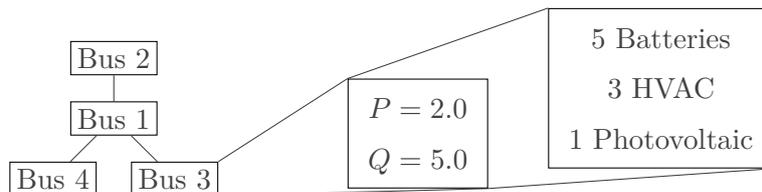


Figure 6: Incorporating data from a collection of DERs into a node in a power flow graph. Units and values of P , Q are arbitrary.

how stochasticity of distributed generation enters into overall power distribution. We now turn to the problem of connecting multiple tools or solvers.

4.4 Connecting Tools

Modeling something as complex as the electrical grid typically involves collaborations among teams who may utilize a variety of tools or implementations, even for the same or similar problems. This creates a need for translation and validation among different solvers. We describe a procedure for accomplishing this task using techniques from the functorial data model, as presented in Subsection 2.1.

Figure 7 provides diagrams describing how to translate between solvers. Consider two solvers for some problem, represented by schemas S and S' , e.g. the schema in Figure 5 and a schema for a solver with a different set of parameters. One can construct a generic solver schema G for the problem, e.g. that in Figure 4, along with queries $Q: S \rightarrow G$ and $Q': S' \rightarrow G$, specifying which information is shared among the generic and specific instances. In this case, one should also define an auxiliary schema A for data which appears in both S and S' but not in G , as well as functors $F: A \rightarrow S$ and $F': A \rightarrow S'$ inserting the data of A into both specific solver schemas.

These constructions give rise to functors between the associated categories of instances, as depicted in Figure 7b. For every instance I of S , we can obtain two instances of S' by applying these functors. These can be combined using a suitable colimit in S' -Inst to get a single instance of S' containing all possible data from S . Such a construction enables one to translate between the inputs, outputs, and parameters for the solver represented by S and the corresponding values for the solver represented by S' .

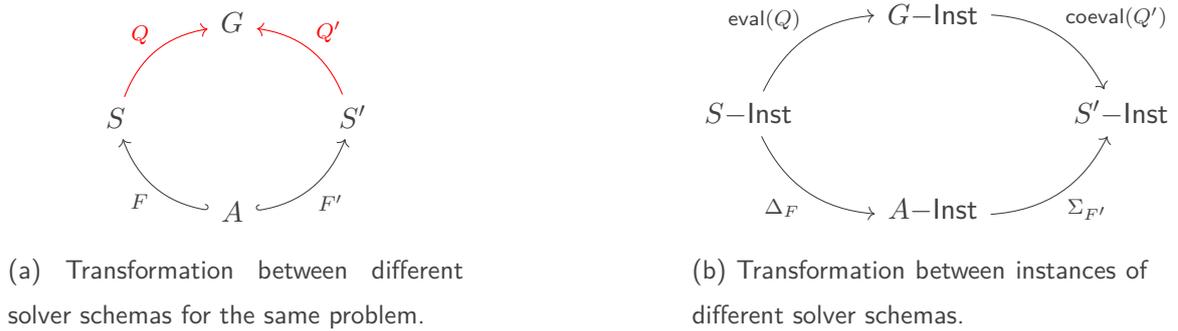


Figure 7: Diagram depicting the transformation of instances for solver schema S to instances for solver schema S' . Black arrows are functors; red arrows are queries.

5 Conclusions and Future Work

This paper provides a window into our efforts to concretize the potential utility of a category theoretic viewpoint for problems dealing with multiple related models and tools in the context of power systems engineering. We saw that techniques and tools from categorical databases can readily be applied to specify and translate among various models, connecting those models to particular analysis tools, as well as connecting various tools themselves.

Further work is required to extend this category-theoretic modeling paradigm to other engineering domains as well as within power systems. What is desired is not a modeling framework which captures the full complexity of the today's grid, but rather a framework which enables the expedient exploration and evaluation of various possible future architectures and pathways to those. The need for such a modeling ecosystem is not unique to power systems.

Of particular relevance for future work in Smart Grid technologies are aspects of control and communication enabled by new devices such as Smart Meters and increased deployment of phasor measurement units (PMUs), devices which measure current, voltage, or phase across the grid. Managing this coupling of an information network with a physical power network presents ample opportunities for applied category theorists.

Lastly, further development of tools for specifying and modeling systems using category theory, e.g. CQL, is essential in terms of engagement with domains. Being able to point practitioners to a system they can get their hands on and play with goes a long way towards arriving at a useful common understanding.

Acknowledgements Source code for the examples discussed can be found at [github:AQL_Powersystems](https://github.com/AQL_Powersystems). The authors would like to thank David Spivak and Ryan Wisnesky of Categorical Informatics Inc. for helpful discussions. This material is based upon work supported by the National Science Foundation under Grant No.1746077. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. JN was supported by the NIST SURF and NIST PREP programs. BP was supported by an NRC Postdoctoral Research Associateship.

Official contribution of the National Institute of Standards and Technology; not subject to copyright in the United States. Certain commercial equipment, instruments, or materials are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose. Parts of this paper may have been presented in technical seminars and included in government publications. Recorded versions of those seminars and copyright free versions of publications are available through the National Institute of Standards and Technology.

References

- [1] Andrey Bernstein, Lorenzo Reyes-Chamorro, Jean-Yves Le Boudec, and Mario Paolone. A composable method for real-time control of active distribution networks with explicit power setpoints. part I: Framework. *Electric Power Systems Research*, 125:254 – 264, 2015. ISSN 0378-7796. DOI: <https://doi.org/10.1016/j.epsr.2015.03.023>. URL <http://www.sciencedirect.com/science/article/pii/S0378779615000905>.
- [2] C. L. T. Borges and R. J. Pinto. Small hydro power plants energy availability modeling for generation reliability evaluation. *IEEE Transactions on Power Systems*, 23(3):1125–1135, Aug 2008. ISSN 0885-8950. DOI: [10.1109/TPWRS.2008.926713](https://doi.org/10.1109/TPWRS.2008.926713).
- [3] Spencer Breiner, Blake Pollard, and Eswaran Subrahmanian. Functorial model management. In *22nd International Conference on Engineering Design*, page To appear. The Design Society, 2019.
- [4] Georgios Chalkiadakis, Valentin Robu, Ramachandra Kota, Alex Rogers, and Nicholas R Jennings. Cooperatives of distributed energy resources for efficient virtual power plants. In *The 10th International Conference on Autonomous Agents and Multiagent Systems–Volume 2*, pages 787–794. International Foundation for Autonomous Agents and Multiagent Systems, 2011.

-
- [5] Zinovy Diskin and Tom Maibaum. Category theory and model-driven engineering: From formal semantics to design patterns and beyond. *Model-Driven Engineering of Information Systems: Principles, Techniques, and Practice*, page 173, 2014.
- [6] Rida T Farouki, Hwan Pyo Moon, and Bahram Ravani. Minkowski geometric algebra of complex sets. *Geometriae Dedicata*, 85(1-3):283-315, 2001.
- [7] Eric O'Shaughnessy Brittany Smith Jeffrey J. Cook, Kristen Ardani and Robert Margolis. Expanding pv value: Lessons learned from utility-led distributed energy resource aggregation in the united states. Technical report, Golden, CO, 2019.
- [8] Michael Johnson, Robert Rosebrugh, and RJ Wood. Entity-relationship-attribute designs and sketches. *Theory and Applications of Categories*, 10(3): 94-112, 2002.
- [9] André Joyal and Ross Street. The geometry of tensor calculus, I. *Advances in Mathematics*, 88(1):55 - 112, 1991. ISSN 0001-8708. DOI: [https://doi.org/10.1016/0001-8708\(91\)90003-P](https://doi.org/10.1016/0001-8708(91)90003-P). URL <http://www.sciencedirect.com/science/article/pii/000187089190003P>.
- [10] Soumya Kundu, Karanjit Kalsi, and Scott Backhaus. Approximating flexibility in distributed energy resources: A geometric approach. In *2018 Power Systems Computation Conference (PSCC)*, pages 1-7. IEEE, 2018.
- [11] Prabha Kundur. *Power System Stability and Control*. McGraw-Hill, 1994.
- [12] A. P. Leite, C. L. T. Borges, and D. M. Falcao. Probabilistic wind farms generation model for reliability studies applied to brazilian sites. *IEEE Transactions on Power Systems*, 21(4):1493-1501, Nov 2006. ISSN 0885-8950. DOI: [10.1109/TPWRS.2006.881160](https://doi.org/10.1109/TPWRS.2006.881160).
- [13] Tom Mens and Pieter Van Gorp. A taxonomy of model transformation. *Electronic Notes in Theoretical Computer Science*, 152:125-142, 2006.
- [14] James R Norris. *Markov chains*. Number 2. Cambridge university press, 1998.
- [15] Farrokh Rahimi and Ali Ipakchi. Demand response as a market resource under the smart grid paradigm. *IEEE Transactions on smart grid*, 1(1):82-88, 2010.
- [16] Lorenzo Reyes-Chamorro, Andrey Bernstein, Jean-Yves Le Boudec, and Mario Paolone. A composable method for real-time control of active dis-

- tribution networks with explicit power setpoints. part II: Implementation and validation. *Electric Power Systems Research*, 125:265 – 280, 2015. ISSN 0378-7796. DOI: <https://doi.org/10.1016/j.epsr.2015.03.022>. URL <http://www.sciencedirect.com/science/article/pii/S0378779615000899>.
- [17] Emily Riehl. *Category Theory in Context*. Dover Publications, 2016. URL <http://www.math.jhu.edu/~eriehl/context.pdf>.
- [18] Mark F. Ruth, Monte S. Lunacek, and Birk Jones. Impacts of using distributed energy resources to reduce peak loads in vermont. 11 2017. DOI: [10.2172/1411137](https://doi.org/10.2172/1411137).
- [19] Patrick Schultz and Ryan Wisnesky. Algebraic data integration. *Journal of Functional Programming*, 27:e24, 2017. DOI: [10.1017/S0956796817000168](https://doi.org/10.1017/S0956796817000168).
- [20] Patrick Schultz, David I. Spivak, Christina Vasilakopoulou, and Ryan Wisnesky. Algebraic Databases. *arXiv e-prints*, art. arXiv:1602.03501, Feb 2016.
- [21] P. Selinger. *A Survey of Graphical Languages for Monoidal Categories*, pages 289–355. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-12821-9. DOI: [10.1007/978-3-642-12821-9_4](https://doi.org/10.1007/978-3-642-12821-9_4). URL https://doi.org/10.1007/978-3-642-12821-9_4.
- [22] David I. Spivak. Functorial data migration. *Information and Computation*, 217:31 – 51, 2012. ISSN 0890-5401. DOI: <https://doi.org/10.1016/j.ic.2012.05.001>. URL <http://www.sciencedirect.com/science/article/pii/S0890540112001010>.
- [23] Michael Stay and L. G. Meredith. Representing operational semantics with enriched Lawvere theories. *arXiv e-prints*, art. arXiv:1704.03080, Apr 2017.
- [24] Perdita Stevens. Generative and transformational techniques in software engineering II. chapter A Landscape of Bidirectional Model Transformations, pages 408–424. Springer-Verlag, Berlin, Heidelberg, 2008. ISBN 978-3-540-88642-6. DOI: [10.1007/978-3-540-88643-3_10](https://doi.org/10.1007/978-3-540-88643-3_10). URL http://dx.doi.org/10.1007/978-3-540-88643-3_10.
- [25] Frank Trollmann and Sahin Albayrak. Extending model to model transformation results from triple graph grammars to multiple models. In Dimitris Kolovos and Manuel Wimmer, editors, *Theory and Practice of Model Transformations*, pages 214–229, Cham, 2015. Springer International Publishing. ISBN 978-3-319-21155-8.
- [26] W. Tushar, S. Huang, C. Yuen, J. A. Zhang, and D. B. Smith. Synthetic gen-

eration of solar states for smart grid: A multiple segment markov chain approach. In *IEEE PES Innovative Smart Grid Technologies, Europe*, pages 1–6, Oct 2014. DOI: [10.1109/ISGTEurope.2014.7028832](https://doi.org/10.1109/ISGTEurope.2014.7028832).

- [27] Ray Daniel Zimmerman, Carlos Edmundo Murillo-Sánchez, and Robert John Thomas. Matpower: Steady-state operations, planning, and analysis tools for power systems research and education. *IEEE Transactions on Power Systems*, 26(1):12–19, Feb 2011. ISSN 0885–8950. DOI: [10.1109/TPWRS.2010.2051168](https://doi.org/10.1109/TPWRS.2010.2051168).