



The Transactive Network Template Metamodel

January 2019

DJ Hammerstrom

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor Battelle Memorial Institute, nor any of their employees, makes **any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights.** Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or Battelle Memorial Institute. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062;
ph: (865) 576-8401
fax: (865) 576-5728
email: reports@adonis.osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
email: orders@ntis.gov <<http://www.ntis.gov/about/form.aspx>>
Online ordering: <http://www.ntis.gov>



This document was printed on recycled paper.

(8/2010)

The Transactive Network Template Metamodel

Click here to enter text.

January 2019

DJ Hammerstrom

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99352

Summary

While transactive energy, which is defined as an allocation of electricity based on dynamically discovered values or prices, has been extensively studied, its uptake and use has been slow. This report describes a tool, the *transactive network template*, which should hasten the creation and uptake of transactive energy networks.

Some basic principles of transactive energy are familiar from existing wholesale electricity markets. Locational prices are calculated today for zones within bulk electric transmission systems. Locational prices differ while accounting for the locational costs of electricity generation and the losses and constraints incurred when electricity is transmitted from generators and distributed to consumers. A transactive energy network might include these transmission zones. However, current research strives to apply transactive energy also in electricity distribution circuits, buildings, and even for individual generating and consuming devices. At the same time, researchers explore how to apply transactive energy in real time during increasingly shorter time intervals.

Automated computational agents become necessary as transactive energy becomes applied to smaller circuit zones and at faster dynamic timescales. A transactive energy network is an example of a multi-agent system. Each zone in the network is represented by its *transactive agent*, which makes decisions for and acts on behalf of a business entity that is responsible for and manages one of the circuit regions. A transactive energy network is also an example of a decentralized, distributed control system. Control decisions and responsibilities are distributed among the network's transactive agents. The transactive agents are independent; that is, there typically is no centralized authority or oversight function. Instead, transactive agents exchange transactive signals and thereby negotiate the prices and quantities of electricity that they will exchange.

Initially, the circuit regions and responsibilities of transactive agents appear to be very dissimilar. Each circuit region may comprise transmission, distribution, or building-level circuits. Each has a unique position and electrical connectivity within the transactive energy network. Each possesses unique assets that either generate or consume electricity, and these (e.g., renewable energy generator, diesel generator, aggregate utility load, building load, space conditioning, refrigerator, etc.) may further differ in their price flexibility and in their strategies for responding to dynamic electricity prices. Given such diversity, an implementer's first inclination might be to start from scratch to define all these devices and to engineer their seemingly unique interactions.

Given that each implementer's perspective may be narrow within a transactive energy network, it is unlikely that uniquely engineered systems would interact well. This is where the transactive network template is applicable. The transactive network template is a metamodel that has been developed to guide implementers as they configure their own transactive agent within a network of such agents. The object-oriented design of the transactive network template provides basic code object types that may be used and extended by implementers to represent each of the assets in their circuit region. These objects further facilitate the transactive agent's necessary computations, which are divided among responsibilities to schedule power usage, balance electric supply and demand, and coordinate the exchange of electricity with the other transactive agents.

This report addresses the conceptual transactive network template design. Implementers are directed to more formal design documents and reference implementations. A Python™-based¹ reference implementation of the transactive network template has been coded, and three implementations have been configured to represent a national laboratory and two university campuses. In the future, the author

¹ Webpage available at <https://www.python.org>.

wishes to make the transactive network template more generally applicable to networks that require more accurate power flow and those employing alternative transactive coordination mechanisms.

Development of the transactive network template is jointly funded by the U.S. Department of Energy (DOE) Energy Efficiency and Renewable Energy and the DOE Office of Electricity. In late 2015, one of the first projects to be funded by the DOE Grid Laboratory Modernization Laboratory Consortium was the Clean Energy and Transactive Campus project, led by Pacific Northwest National Laboratory. DOE funds were matched by an investment by the Washington Department of Commerce through its Clean Energy Fund. The transactive network template was developed to guide the implementation of transactive energy networks within this project's scope.

Glossary and Acronyms

<i>local asset</i>	A device or system that generates or consumes electricity and that is managed by a <i>transactive agent</i> . An object class within the <i>transactive network template</i> . A <i>transactive agent</i> knows all states and strategies of its <i>local assets</i> . A <i>transactive agent</i> schedules its <i>local assets</i> , but it does not transact with them.
<i>locational marginal price</i>	Formal definitions exist in wholesale electricity markets. More generally, the change in cost that accompanies a change in electricity supply at a given location and time.
<i>transactive agent</i>	An entity that manages a circuit region using an implementation of the <i>transactive network template</i> . A <i>transactive agent</i> must negotiate with other <i>transactive agents</i> to coordinate energy flow and value within its <i>transactive energy network</i> .
<i>transactive energy network</i>	A set of <i>transactive agents</i> , plus the allowed transaction pathways between the <i>transactive agents</i> .
<i>transactive neighbor</i>	Each member of a pair of <i>transactive agents</i> that transact with one another by exchanging <i>transactive signals</i> .
<i>transactive network template</i>	An abstracted, object-oriented metamodel of one <i>transactive agent</i> 's perspective within a multi-agent <i>transactive energy network</i> . An extensible set of base object classes that was developed using the <i>Unified Modeling Language</i> [®] . A reference implementation of the transactive network template base classes has been instantiated in Python TM ¹ for use with the VOLTTRON TM ² platform.
<i>transactive signal</i>	The information that <i>transactive neighbors</i> exchange. A set of records, each of which represents a pairing of price and quantity within a forecast time interval.
UML [®] <i>Unified Modeling Language</i>	A standard visual modeling language intended to be used for modeling business and similar processes, analysis, design, and implementation of software-based systems. ³ The standard is maintained by the Object Modeling Group [®] .

¹ Webpage available at <https://www.python.org>.

² Webpage available at <https://volttron.org/>.

³ Webpage available at <https://www.uml-diagrams.org>.

Contents

Summary.....	iii
Glossary and Acronyms.....	v
1.0 Introduction.....	1
2.0 The Transactive Network Template Metamodel.....	3
2.1 Base Transactive Network Template Code Classes	3
2.2 Scheduling, Balancing, and Coordination Objectives	4
2.2.1 The Balancing Responsibility.....	4
2.2.2 The Scheduling Responsibility.....	9
2.2.3 The Coordination Responsibility.....	11
2.3 What It Means to <i>Configure</i> a Transactive Network Template Code Implementation	15
2.4 What It Means to <i>Specialize</i> a Class	15
3.0 Suggested Future Work	16
3.1 Power Flow Improvements.....	16
3.2 Facilitate Different Alternative Price-Discovery Mechanisms	17
3.3 Support of Sequential Correction Markets.....	18
3.4 Support of Financial Transactions.....	18

Figures

1 Simplified Representation of a Transactive Agent’s Transactive Network Template Code Implementation and Its Interfaces to the Transactive Agent’s Local Assets and Transactive Neighbors	4
2 A Simple Example Clearing between a Flexible Supplier and Inflexible Consumer	8

Tables

1 Important Properties and Responsibilities of Transactive Network Template Base Class Objects.....	5
---	---

1.0 Introduction

The *transactive network template* is an extensible set of base object classes that was developed using the Unified Modeling Language™ (UML).¹ The transactive network template facilitates the configuration and operation of one agent within a cyber-physical network of such agents. It is designed to facilitate decentralized transactive energy systems, which automate and coordinate decentralized control decisions of distributed devices that generate or consume electricity.

Development of the transactive network template is jointly funded by the U.S. Department of Energy (DOE) Energy Efficiency and Renewable Energy and the DOE Office of Electricity. In late 2015, one of the first projects to be funded by the DOE Grid Laboratory Modernization Laboratory Consortium was the Clean Energy and Transactive Campus project, led by Pacific Northwest National Laboratory. DOE funds were matched by an investment by the Washington Department of Commerce through its Clean Energy Fund. The transactive network template was developed to guide the implementation of transactive energy networks within this project's scope.

Matlab^{®2} and Python^{®3} reference implementations of the transactive network template were coded. The Matlab implementation is useful for research exploration, but it is not suitable for field use. The Python reference implementation has been implemented on various VOLTTRON™⁴ communication platforms for use in actual networks. Furthermore, the Python reference implementation was used to configure three transactive energy network implementations—one representing a transactive energy network comprised of buildings on the Pacific Northwest National Laboratory campus and their upstream campus, municipality, and wholesale electricity suppliers; and two representing two U.S. university campuses.⁵

The word *template* highlights the ability of the transactive network template to guide and facilitate specific agent implementations. Its object-oriented design enforces a degree of standardization, and its classes provide the standard properties that will be needed by an agent. Furthermore, needed behavioral methods are enforced to standardize inter-agent transactions and integration of the agent's local assets. At the same time, extensibility is supported for the assets themselves, which often possess additional unique properties and energy behaviors. Extensibility is quite naturally supported when using object-oriented code design. Libraries of asset models (e.g., for specific building loads and demand-responsive assets) should evolve by inheritance and extension of the base transactive network template object classes. The base classes themselves should not be casually modified.

This report is intended for a nontechnical reader, not the coder or implementer. It introduces concepts and features of the transactive network template, but it does not provide enough detail to create another valid reference implementation. It is important for the reader to understand why the transactive network template is a *metamodel*, but the reader is not expected within this report to interpret the UML diagramming which allowed its design as an object-oriented metamodel. If still more detail is needed, contact Donald J. Hammerstrom⁶ regarding the design document or Hung Ngo⁷ regarding the Python reference implementation code.

¹ Webpage available at <https://www.uml-diagrams.org>.

² Webpage available at <https://www.mathworks.com/products/matlab.html>.

³ Webpage available at <https://www.python.org>.

⁴ Webpage available at <https://volttron.org/>.

⁵ Katipamula S, RG Lutes, S Huang, J Lian, H Ngo, and DJ Hammerstrom. 2019. *Coordination of Behind-the-Meter Distributed Energy Resources for Transactive Grid Services: Multi-Building*. PNNL-XXXXX, Pacific Northwest National Laboratory, Richland, Washington.

⁶ Donald.Hammerstrom@pnnl.gov

⁷ Hung.Ngo@pnnl.gov

Implementations of the transactive network template are intended to facilitate a single agent's perspective within a multi-agent transactive network. That agent will be referred to in this report simply as "the agent" or "this agent." When necessary, references to "other network agents" or "another network agent" or "neighboring agent" should be understood to refer to other members of the multi-agent network besides the one central to the given implementation. Terse, precise class names were used in the transactive network template, but more readable names and descriptions have been used in this report (e.g., a *LocalAsset* object is referred to as a "local asset object").

The structural and behavioral aspects of the transactive network template are addressed in Section 2, and further improvements to the transactive network template are suggested and discussed in Section 3.

2.0 The Transactive Network Template Metamodel

The transactive network template is a *metamodel* in that it is a model of models. Its object-oriented UML design may guide reference implementations that may use different software code languages. Regardless, every reference implementation should reproduce the same transactive network template base classes and base class behaviors. In principle, transactive agents residing in the same transactive network may choose entirely different reference implementations of the transactive network template, and these agents should still properly exchange transactive signals and interact.¹

Object-oriented designs are separable into their structural and behavioral elements. Section 2.1 introduces base classes of the transactive network template, thereby providing a structural overview of the structural elements that are available to model an agent's assets and position within a transactive network. Section 2.2 introduces the most important behavioral responsibilities of the transactive agent and how those responsibilities are allocated among the available transactive network template classes. The three fundamental computational responsibilities of a transactive agent are to 1) to balance electrical power in the circuit region that is managed by the agent, 2) schedule the power to be generated or consumed by the agent's local assets, which decisions may be price-responsive, and 3) conduct transactions and coordinate electricity exchanges with other agents.

Sections 2.3 and 2.4 address what it means to configure and extend a transactive network template code implementation respectively.

2.1 Base Transactive Network Template Code Classes

This section introduces the structure of the transactive network template by introducing its base classes. These classes are the structural elements available to design, configure, and operate a transactive network template implementation. Objects must be instantiated from these base classes (or from classes whose parentage can be tracked back to these base classes) from the perspective of a single transactive agent and its electric circuit region that it represents.

A transactive network template implementation provides useful object classes that one transactive agent may configure, specialize, and use to plan and manage electricity supply usage in the circuit region for which it is responsible. Three of the transactive network template's most important code classes are shown in Figure 1. The *market class* manages the transactive agent's balancing of electricity supply and demand. The *local asset class* interfaces with and schedules the devices and systems in the circuit region that the transactive agent manages. The *neighbor class* coordinates the transactions and manages the exchange of transactive signals with neighboring transactive agents. The neighbor and local asset classes may be specialized and instantiated as many times as is necessary to represent all the transactive agent's assets and transactive neighbors.

The transactive network template features more object classes than the important ones featured in Figure 1. High-level information about all the transactive network template base classes is summarized in Table 1. The first column gives both the base class name (bold, italicized) and its brief description. The second column lists properties that the base class manages on behalf of the transactive agent, and the third column lists the classes' most important behavioral responsibilities.

¹ More precisely, the transactive records should interoperate at the business and syntactic levels. The transactive network template is agnostic about interoperability in physical communication layers. Neighboring agents may have to negotiate their choice of communication carrier and choose from available low-level communication protocols.



Figure 1. Simplified Representation of a Transactive Agent’s Transactive Network Template Code Implementation and Its Interfaces to the Transactive Agent’s Local Assets and Transactive Neighbors

2.2 Scheduling, Balancing, and Coordination Objectives

There are three very important computational responsibilities managed by transactive agents that implement the transactive network template: 1) balancing, 2) scheduling, and 3) coordination. The transactive network template has been designed to make these three computations as separable and independently achievable as possible.

2.2.1 The Balancing Responsibility

Many readers will possess conceptual understanding of the market principles that are central to the balancing objective. Given updated power schedules and price flexibilities of all the agent’s local assets and neighbors, an agent’s market object calculates an electricity price that balances supply and demand among all electricity entering and exiting the agent’s circuit region.

The agent’s circuit region is treated as a “copper plate,” which means it possesses undifferentiated electrical circuit properties and incurs no transport losses within the circuit region. There may be only one voltage in the circuit region. While exchanges with neighbor objects may be modified to reflect electricity that is lost upon importing electricity into the circuit region, local assets reside in the circuit region and typically will not incur transport losses.¹ The current transactive network template version addresses only real electric power transactions and the balancing of real power.²

¹ These “copper-plate” principles are intentional and should not be violated. Electricity must have one unit price across the agent’s entire circuit region at any given time. If an implementer feels compelled to assign multiple electricity prices within one agent’s circuit region, the regions should be separated and granted their own agents and transactive network template implementations.

² Future versions should address reactive power and voltage management, which would require successively detailed calculations of both real and reactive power generation and consumption within the agent’s circuit region and complex power transport between agents and their circuit neighbors. Effects like transport losses can be estimated until such new transactive network template versions can be completed. These future versions are addressed in Section 3.0.

Table 1. Important Properties and Responsibilities of Transactive Network Template Base Class Objects

Class Object	Important Properties	Responsibilities
<i>myTransactiveNode</i> —The agent’s transactive node. It represents an agent’s perspective from within its circuit region.	<ul style="list-style-type: none"> Lists of information source, local asset, market, meter, and neighbor objects that the asset must interact with. 	<ul style="list-style-type: none"> None.
<i>LocalAsset</i> and <i>LocalAssetModel</i> ^(a) —A local device or system that must be scheduled by the transactive agent—a generation or load device that lies within the agent’s circuit region and is “owned by” the asset. The transactive agent knows the asset’s entire status and strategy.	<ul style="list-style-type: none"> Cost parameters for calculating production costs. Default and active vertices for representing the asset’s flexibility to the locational price. Lists of meter and information sources. The object’s name and description. The asset’s default, minimum, maximum, and scheduled powers. The asset’s production, total production, dual, and total dual costs. 	<ul style="list-style-type: none"> Given forward electricity prices, an asset schedules its own power and its flexibility to change its electric power generation or consumption in response to changes in those forward prices. Given the asset’s scheduled powers, the asset updates its production and dual costs, which are used by the market object to determine convergence of the transactive agent’s balancing objective.
<i>Neighbor</i> and <i>NeighborModel</i> —Locations outside the agent’s circuit region with which the agent may exchange electricity. Transactive neighbor objects are further managed by other transactive agents of the transactive network and expect to exchange transactive records with this agent.	<ul style="list-style-type: none"> The neighbor’s default, minimum, maximum, and scheduled powers. Cost parameters that may be used to calculate production costs.^(b) Default and active vertices that are used to represent neighbor flexibility on a marginal supply or demand curve. Demand-charge parameters. Lists of meter and information sources. The object’s name and description. Production, total production, dual, and total dual costs. Ready, sent, and received transactive records. Boolean indicator stating whether neighbor is transactive or not. Transport loss factor that may be used to estimate transport losses for electricity imported from the neighbor. 	<ul style="list-style-type: none"> Given a series of forward electricity prices, the <i>Neighbor</i> schedules its own power and its flexibility to changes in those forward prices. Given the neighbor’s scheduled powers and forward prices, the neighbor updates its production and dual costs. For every neighbor object, the agent must prepare, send, receive, and check for convergence among transactive record signals. If demand charges are in play for the neighbor object, it must update the thresholds and impacts of the demand charges.

Class Object	Important Properties	Responsibilities
Market	<ul style="list-style-type: none"> Aggregate flexibility is stored as a list of aggregate active vertices. Aggregated generation total generation, demand, total demand, and net powers. Aggregated production, total production, dual, and total dual costs. Convergence criterion threshold concerning the agent's balancing objective. Convergence status. Default and actual marginal electricity prices. Market forecast horizon and clearing schedule. Time interval duration. 	<ul style="list-style-type: none"> The market object must balance electric supply and demand for the agent, which includes these following responsibilities. Initiate updating of and sum local assets' and neighbors' production and dual costs. Initiate updating of and sum of local assets' and neighbors' powers and flexibility scheduling.
TimeInterval —A time interval object. ^(c)	<ul style="list-style-type: none"> Interval activity, market convergence, market state, and reconciliation statuses. Market object in which the time interval is relevant. The object's name. The interval's starting, duration, activation, and clearing times in the given market object, as well as its calculation timestamp. 	<ul style="list-style-type: none"> Update market state. A construction method exists to enforce class structure.
IntervalValue —Measured or calculated data that belong within their specified time interval. For example, forecasted price and power data must have their respective time interval specified.	<ul style="list-style-type: none"> Market object in which the value is relevant. Object name identifier. Source class and object that created value. The value and its measurement type and units of measure. Time interval in which the value resides. 	<ul style="list-style-type: none"> A construction method exists to enforce class structure.
TransactiveRecord —transactive record. A set of these records constitute a signal between this agent and one of its transactive neighbors.	<ul style="list-style-type: none"> Tuple of cost, marginal price, and power, and the time interval in which the tuple is relevant.^(d) Indicator whether the record represents scheduled power or an inflection point on a piecewise linear supply or demand curve. 	<ul style="list-style-type: none"> A construction method exists to enforce class structure.
Vertex —An inflection point between lines of a piecewise linear marginal supply or demand curve.	<ul style="list-style-type: none"> Tuple of cost, marginal price, and power. 	<ul style="list-style-type: none"> A construction method exists to enforce class structure.
MarketState —Enumeration of possible market states.	<ul style="list-style-type: none"> The set {Inactive, Explore, Tender, Transaction, Delivery, Publish, and Expired} 	<ul style="list-style-type: none"> None.

Class Object	Important Properties	Responsibilities
MeterPoint —A meter source of data. ^(e)	<ul style="list-style-type: none"> • Meter description and name. • Measurement value and type and unit of measure. • Sample time and interval and next scheduled sample time. • Store time, storage interval, and next scheduled repository store time. 	<ul style="list-style-type: none"> • Read a meter. • Store meter data.
InformationServiceModel — Information service or, more generally, a source of information other than a simple meter. ^(f)	<ul style="list-style-type: none"> • Information source (e.g., service and its location). • Information type and units of measure. • Object name. • Sample time and duration, and next scheduled sample time. 	<ul style="list-style-type: none"> • Update information (e.g., forecast outdoor temperature from a weather forecasting service).

^(a) Two classes of types object and model were used for various base classes of the transactive network template. The intention was to group object properties that reside with objects in static time and behavioral properties and methods that reside modeled with objects in predicted time series. There is some value to this approach, but the distinction will be mostly ignored within this report.

^(b) Fueled generator cost functions are the basis for most theory underlying wholesale electricity markets. The transactive network template therefore facilitates calculation of production costs using a quadratic function of power. Interestingly, these conventional cost functions are not particularly useful for determining the production costs of other distributed generation resources.

^(c) Time interval objects, once instantiated, remain affixed to their delivery time period. The time interval object can therefore keep track of a market's status, which facilitates potentially rich market timing practices. The transactive network template should be resilient to missed calculations and system down times because of the persistence of time interval objects.

^(d) Many additional tuple elements were defined for a transactive record to support anticipated future functionality. Unimplemented, untested features will not be discussed in this report.

^(e) As for *InformationServiceModel*, reference implementations are tending to ignore this class. It was hoped that this class would facilitate platform independence.

^(f) The reference implementations largely ignore this class. Access to some information sources will be found to have been provided by a communication platform, as was the case with VOLTTRON. It was hoped that this class would facilitate platform independence.

All suppliers, consumers, importers, and exporters of electricity are treated similarly and symmetrically by the transactive network template. A consistent sign convention is enforced from the perspective of the agent and its circuit region that the agent manages. Power and electricity entering into the agent's circuit region via generation or importation are assigned positive values; power and electricity exiting the agent's circuit region via consumption or exportation have negative values. Bidirectional power flows and energy storage assets are permitted and supported. The transactive network template allows an electricity consumer to become a generator or an exporter to become an importer from one time interval to the next. An important consequence of this sign convention is that, from the agent's perspective, electric power balance has been achieved if the sum of all electricity generation, consumption, importation, and exportation is zero.

The balancing process is initiated upon the agent's market object inviting all its neighbor and local asset objects to update their schedules, price flexibilities, and production and dual costs. These scheduling computations by the transactive network template neighbor and asset objects are precisely the responsibilities that are to be discussed in Section 2.2.2. The market object sums the objects' responses to determine net power balance, net available price flexibility, and various costs that indicate whether the

balancing process has converged.¹ The transactive network template design is intended to eventually support several alternative coordination mechanisms and solution methods. To date, it supports two alternative solution methods: Method 1 conducts a sub-gradient search, in which each time interval's price is nudged up or down based on the magnitude and sign of the calculated duality gap. This first method requires many iterations, but it becomes necessary when objects cannot, or choose not to, reveal the flexibility of their scheduled powers to price changes. The balancing process ends when the duality gap has been driven to a suitably small magnitude. The simpler, preferred Method 2 requires many fewer iterations. If all transactive network template neighbor and asset objects reply to their market object with accurate, piecewise linear representations of their price flexibility for each active time interval, then the clearing price may be accurately determined through interpolation, thus requiring few if any iterations. The duality gap still is used by Method 2 to indicate convergence, but very few iterations were needed for the simple reference implementations to date.

The authors strongly advocate for the simpler Method 2. One of its assumptions is that all objects' production (and consumption) costs are represented by quadratic, monotonically increasing cost functions. An implication of this common assumption is that the derivative of the quadratic cost function—its marginal supply or demand curve—is affine. Any power function comparable to marginal prices is therefore piecewise linear, as demonstrated by Figure 2, which shows a cleared balancing between a price-responsive supplier and an inflexible consumer.

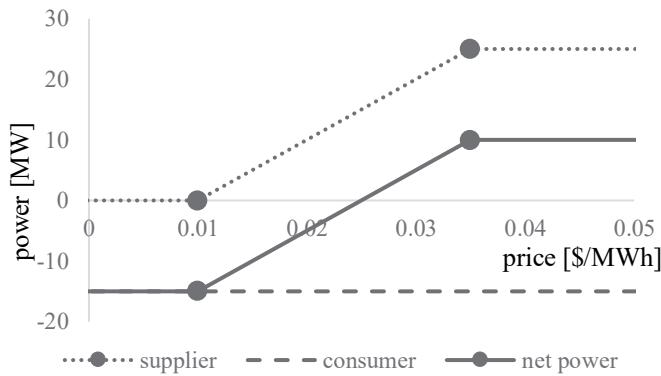


Figure 2. A Simple Example Clearing between a Flexible Supplier and Inflexible Consumer

The price-sensitive supplier's offer is represented by three line segments: It will never produce electricity at any price below \$0.01/MWh. It will produce a linearly increasing power between that price and \$0.035/MWh. Finally, it can produce no more than 25 MW at any price higher than that. In this case, only two inflection points, or *vertices*, were needed to represent a relatively sophisticated price-responsive production offer from the supplier.

Curves' tails are always presumed to extend horizontally from the smallest priced vertex to negative infinity and from the greatest priced vertex to positive infinity. The price-inflexible consumer in Figure 2 is represented by a single horizontal line at -15 MW. This entire bid from the consumer may be represented by the single vertex $(\infty, -15)$.

Net power is calculated by adding powers at each marginal price. In fact, if the curves are all piecewise linear, only power at vertex marginal prices must be added, and the power at all other marginal prices

¹ More precisely, the magnitude of the *duality gap*, which is a difference between *primal* and *dual* costs, indicates whether the agent has converged upon a satisfactory balance of supply and demand throughout a set of forecast time intervals.

may be found by interpolation. The corresponding clearing price occurs where the net power curve (again piecewise linear) is zero, meaning that supply and demand are of equal magnitude (i.e., balanced).

Incidentally, the transactive network template allows and supports negative marginal prices. A negative marginal price indicates that balancing clears to the left of \$0/MWh, which requires that a supplier was willing to reduce its production, or a consumer was willing to increase its consumption, as marginal price became negative. While marginal price remains negative, suppliers must pay to produce, and consumers must be paid to consume electricity.

2.2.1.1 Startup and Future Balancing Issues

System startup of transactive network template processes creates interesting challenges for the balancing process. The market object must supply default marginal prices to the agent's local asset and neighbor objects, which any price-responsive object will require if it is to schedule its power. While this might seem to be an issue only during system startup, new future time interval objects are constantly being spawned as the system marches through time, and a default marginal price must be assigned to these new time intervals as well to jumpstart the balancing of power in the new time intervals.

Base transactive network template classes may possess additional class properties and methods that are not discussed in this report. In most cases, these features were included in the transactive network template to support anticipated future functionality. We list some anticipated functionality here but warn the reader that these capabilities are not tested and may be incomplete:

- *Reserve margins.* Given an increasing interest in grid resilience, properties have been proposed to help keep track of aggregate system reserve margin. Unused, super marginal production capacity (and potentially unengaged demand reduction, as well) might be claimed as reserve margin. The available aggregate reserve margin might indicate the transactive network's resilience. Alternatively, a requirement might be stated for a minimum allowable reserve margin, and the market class clearing process might be extended to account for the achievement and cost of this requirement.
- *Asset engagement.* Wholesale electricity markets perform unit commitment to preplan whether large generators need be engaged (ready) or not. This practice is particularly important for assets that take a long time to start up and shut down or that incur costs upon doing so.
- *Fixed and fixed, avoidable costs.* Fixed production costs are not typically used for unit commitment and dispatch, but fixed costs comprise a substantial fraction of final electricity bills. And some wholesale electricity markets are finding ways to re-compensate producers for fixed, avoidable expenses (e.g., startup cost) they might incur. For these and other reasons, transactive systems may need to harmonize discrepancies and unfairness due to fixed and fixed, avoidable costs.

2.2.2 The Scheduling Responsibility

Given a series of forward prices, each neighbor and local asset object must be able to 1) predict its power generation or consumption, 2) calculate its predicted flexibility to change its schedule in response to changes of said forward prices, and 3) calculate its various production and dual costs, which may include the costs of both electricity and utility. These responsibilities are referred to here as *scheduling*. Each local asset and neighbor object should be able to independently schedule itself. Each object acts in its self-interest on behalf of its transactive agent.

The scheduling process is initiated when the local asset or neighbor objects are invited by the transactive network template market object to schedule themselves. The scheduling processes differ slightly for neighbor and local asset objects, as will be addressed in the next two subsections.

2.2.2.1 Scheduling of a Neighbor Object

Because the neighbor object represents a neighboring agent and its remote circuit region, this agent (meaning the one central to the transactive network template) knows little about the remote circuit region and its motivations and therefore uses a more standardized, hard-coded approach in its scheduling of the neighbor object. The neighbor object maintains a copy of the last supply or demand curve that was received from the neighboring agent in the form of a transactive signal. These curves are represented by the inflection vertices, as has already been discussed in conjunction with Figure 2. Therefore, the scheduled power is simply the power at which the neighbor object's saved supply or demand curve intersects the current marginal price in its respective time interval. The neighbor's price flexibility is precisely represented by the neighbor object's saved supply or demand curve that it received also from the neighbor transactive agent via a received transactive signal. The production cost (or its equivalent gross consumer surplus), excluding a constant term, may be calculated by integrating the neighbor object's saved supply or demand curve over the object's viable power from its minimum production (maximum consumption) to maximum production (minimum consumption). The dual cost is equal to the calculated production cost (gross consumer surplus), less the energy income (outflow).

2.2.2.2 Scheduling a Local Asset Object

During the scheduling process for local asset objects, the market object expects the same calculated results as from neighbor objects, but the scheduling of local asset objects may be much more diverse and complex and highly specialized. This complexity can be accommodated for the local asset objects, however, because the status of a local asset object is always fully transparent to its agent. Any strategy of a local agent is fully known by its agent and should exist to serve the interests of the agent, as well.

Consider a battery energy storage site as an example local asset represented by its corresponding local asset object. An agent's owner might pose a strategy to optimize the arbitrage value of the energy to be stored into batteries (negative power and energy, by the transactive network template sign convention) and released back to the electric power grid (positive power and energy). This is certainly feasible to do given a series of forward electricity prices. However, the optimization strategy must consider the site's limited energy storage capacity, the inverters' charge and discharge power capacities, and the batteries' current state of charge. This optimization is very challenging, but the optimization strategy might still be configured to suit not only this example, but also the needs of other battery energy storage owners who have similar basic objectives. More sophisticated battery owners might further value battery lifetime, backup reserve capacity, maintenance periods when battery availability is to be limited, and so on.¹ The potential objectives are unbounded. Therefore, many local asset battery energy storage objects will become further specialized to address permutations of more and more operational objectives, resulting in increasingly richer optimization strategies. For these reasons, the transactive network template establishes an interface by which local asset objects must respond their power generation or consumption schedules, but the transactive network template must not formalize or standardize the means by which the optimal power schedule is calculated.²

We continue the battery energy storage example and consider its responsibility to calculate its price flexibility. Assume that an optimal power schedule has been found. Further define *residual flexibility* as a

¹ Upon including valued objectives in addition to electricity cost and income, one is now optimizing a *utility cost* function.

² In fact, many empirical and simplified heuristic methods and decision functions have evolved for predicting transactive power schedules from prices. Analytical optimization should be preferred for power scheduling when explicit forward prices exist, as is the case today for the transactive network template. Simplified approaches should strive to approximate the results of a true optimization strategy.

trajectory on the marginal price plane if one time interval's price were to be perturbed while leaving all other forward prices unchanged. The point at the scheduled power value and current locational electricity price is necessarily a member of the residual flexibility curve. All the other points on the trajectory may be found while using the same optimization strategy as was used to find the scheduled power.

The opportunities represented by the residual flexibility curve are therefore respectful of the opportunities the asset has in the other time intervals because the multi-period optimization principals are retained unchanged. It is important for implementers to understand this ideal approach (using perturbation to determine residual flexibility) even if shortcuts are taken.¹

Regardless how residual flexibility is calculated, it should become represented by its piecewise linear supply or demand curve. Doing so takes advantage of concise storage and simple calculations, as was discussed in conjunction with Figure 2.

The final scheduling objective for the local asset object, to calculate production cost (gross consumer surplus) and dual cost, proceed as for neighbor objects as discussed earlier.

Much as the transactive network template market object provided default prices to facilitate bootstrap startup of the transactive network template balancing process, both local asset and neighbor objects must supply a default scheduled power and default active vertices to ensure startup and resiliency of the scheduling processes. If an agent's transactive network template implementation is to successfully start up, then there must exist a feasible initial solution to the balancing computation—there must exist a price at which summed supply and load balance one another.

These default values are particularly needed by a neighbor object, which lacks knowledge of the neighboring agent's power needs (or offers) and price flexibility until that neighbor agent chooses to reveal such information by finally sending its transactive signal. Local asset objects are less dependent on these defaults during bootstrap startup, but the defaults may still be used should scheduling calculations fail.

2.2.3 The Coordination Responsibility

Pairs of transactive neighbor agents transact; that is, they exchange transactive signals. In doing so, each agent plays its role in the coordination of energy allocation throughout the entire transactive network of decentralized, independent transactive agents. Neighboring transactive agents should be independent. Each has its own unique local assets and its own unique position within the transactive network. A neighboring transactive agent cannot even be expected to have been derived from the same transactive network template reference implementation or to have been coded using the same computer language. A transactive agent can decide when to send its own computed transactive signal to one of its neighboring agents, but it cannot insist and control precisely when that neighboring transactive agent shall reply with its (i.e., the neighboring agent's) transactive signal. The transactive network template coordination and signaling processes must therefore be flexible to accommodate other agents' independence.

A *transactive record* pairs a forecast time interval with a record number, price, and electricity quantity.² Neighboring transactive agents must agree on the format of a transactive record so that exchanged records may be accurately interpreted by both neighbors. Better yet, an entire transactive network should agree upon and enforce a standard transactive record format (along with compliant transactive network template

¹ Again, if optimization principles were not applied while calculating a power schedule, then there is no strong support for calculating price flexibility. The local asset's intertemporal constraints may be lost and violated.

² The transactive network template transactive record class currently includes other properties such as reactive power and voltage that will be needed for future transactive network template capabilities and versions.

reference implementations) as a condition of joining a transactive energy network. The transactive network template specifies content information and structure of transactive records, but it is agnostic concerning low-level communication protocols that must be negotiated between transactive neighbors.

At least one transactive record must be sent for each active time interval, thus revealing the agent's aggregate electric power that, according to its computations, is scheduled to be exchanged. This record is assigned as "Record 0". When only this one single transactive record is sent, the sender is saying it can offer no flexibility to change its scheduled power as a function of the time interval's electricity price.

If additional transactive records are sent concerning an active time interval, these records are numbered successively using integers $\{1, 2, \dots\}$. These additional transactive records represent the agent's residual flexibility—inflection points ("active vertices") of the agent's supply or demand curve in the active time interval. At least two inflection points are necessary to represent residual flexibility.¹

Neighboring transactive agents must agree which time intervals are eligible for transactions and should therefore be represented among the transactive records that are sent and received.² The set of transactive records for all the active time intervals is the transactive record *signal*.

If the transactive signals between neighbors can be represented by relatively few transactive records, each representing an inflection point of a piecewise linear supply or demand curve, then the transactive signals remain small and should not require much communication bandwidth. The transactive network template is agnostic concerning the choice of communication carrier. An agent may be, but is not necessarily, internet connected. Future transactive network template versions should not specify or require any one single communication protocol.

The coordination process is driven by the agent's decision to send its transactive signal to a neighboring transactive agent. It should do so upon recognizing either of two events: 1) local conditions have significantly changed since a transactive signal was last sent or 2) the agent and its neighboring agent disagree concerning the price or average electric power that is to be exchanged.

To determine these events, the agent saves and compares three versions of the transactive signal: 1) the last transactive record signal that was sent, in fact, by this transactive agent to the neighboring transactive agent, 2) the current transactive signal that is up-to-date and ready to be sent to the neighboring transactive agent, but has not yet been sent, and 3) a copy of the most recent transactive record signal received by this transactive agent from the neighboring transactive agent. The transactive agent should send its transactive signal if the prepared and sent transactive record signals differ by more than a configurable threshold magnitude. The agent should also resend its transactive signal if the prepared and received transactive signals differ by more than another configurable threshold magnitude, providing the neighbor's transactive signal has been received since one was last sent. The transactive agent believes coordination between itself and its neighboring agents has converged while neither event compels it to resend its transactive record signal.³

¹ Specifically, this means there may be sets of transactive records numbered $\{\{0\}; \{0, 1, 2\}; \{0, 1, 2, 3\}; \dots\}$, but never $\{0, 1\}$, for a given active time interval.

² This determination concerning *active* time intervals and other attributes and timing of the transactions are specified by the transactive network template market object. Neighboring agents must therefore agree on transaction rules as specified by their respective transactive network template objects.

³ This fully event-driven design is very flexible and forgiving. The system can ride through temporary communication outages, but it cannot distinguish satisfied neighbors from those experiencing failed communications. Therefore, a recommended practice is for an agent to send a transactive signal at least once during each market time interval. Future transactive network template versions should also discount the reliability of stale transactive signals that were not updated when expected.

The transactive network template coordination process might have built upon conventional bilateral auctions in the formulation of its base market class, but doing so would probably limit the types of market interactions that could be supported by the transactive network template. Bilateral auctions in multiple forecast time intervals require fairly strict timing. An auction process typically aggregates supply and demand participants, and all participants must therefore be queried and updated before market clearing can commence. Instead, the transactive network template market object is based upon an exchange of *net* supply or demand curves between any two neighboring agents. The *net* supply or demand curve is the sum net flexibility of all of an agent's neighbor and local asset objects, *excluding the transactive neighbor agent to which the transactive record signal pertains*. The principle may be exemplified using Figure 2, which shows a supply, demand, and summed net curve. If the supply curve is that revealed by a neighboring agent that supplies this agent, then its transactive signal would exclude the supply curve, leaving only the constant demand curve to be included in the transactive record signal. The points are that 1) coordination may occur between two transactive neighbors independently from any aggregation and 2) the timing of signal exchanges is not necessarily limited to prescribed market periods or heartbeat dependencies.

If an agent and a neighboring agent are to initiate successful transactions, each must configure generous default representations of the other's power capacity range. Each neighbor has an assigned minimum and maximum power capacity that represents such capacity. The electric power received from or sent to a neighbor may be constrained either by the abilities of the two neighbors to generate and consume electricity or by the capacity of the electricity transport elements (e.g., conductor, transmission line, distribution line, transformer, fuse, breaker, etc.) between the two neighbors. The capacity range may be narrowed to powers either above or below zero if a neighbor agent is anticipated to always be an electricity supplier or consumer.

Each agent may compute a still narrower power range to represent its residual flexibility range, wherein it could be enticed to operate given correspondingly enticing prices. The agent's residual flexibility may extend to either or both the capacity constraints. On the other hand, if the transactive agent truly has no price flexibility, its residual flexibility becomes a single scheduled power within the allowed power capacity range.

The relationships of these various hard and soft power constraints are shown in Eq. (1). Bars below and above average powers p represent minima and maxima, respectively, of the indicated ranges.¹ Transactive records should represent the scheduled power and soft price flexibility range, if any. An agent should offer as much price flexibility as it can (and will) via its transactive records. A transactive record between two transactive neighbor agents should never be computed to lie outside the hard capacity constraint range that they share. The logic of this paragraph defines and enforces the impacts of price-responsiveness and capacity constraints during the transactive network template's coordination of power exchange between neighboring transactive agents.

$$\underbrace{p_{\text{capacity}} \leq p_{\text{flexibility}} \leq p_{\text{scheduled}} \leq \bar{p}_{\text{flexibility}} \leq \bar{p}_{\text{capacity}}}_{\text{price flexibility range}} \quad (1)$$

¹ Observe that the transactive network template sign convention applies here. The minimum p_{capacity} for a neighbor object that consumes electricity from this agent is a negative number that represents its *maximum* average electric power consumption during the time interval, a negative number.

Discussion has thus far avoided the interpretation of the agent's electricity price. For implementations that support only a shallow network (e.g., utilities bidding their aggregate local asset flexibilities into a wholesale electricity market), the interpretation might not matter much. Local asset objects will schedule themselves given any forward price schedule, regardless of its interpretation. However, for deep, greatly decentralized networks, the interpretation of the price must be consistent with the market process and determines whether assets are meaningfully coordinated among the many levels of the transactive network. Marginal price is the preferred interpretation of a transactive agent's price. It is the locational marginal price that meaningfully compares resource opportunities and determines which distributed resources should be dispatched.¹

Whereas local assets were said to reside on a lossless copper plate representation of the agent's circuit region, neighbor objects are remote, do not lie on the lossless copper plate, and should account for any losses that are incurred as electricity is imported into the agent's circuit region. A parameter of the neighbor class allows an implementer to configure the full-load percentage loss when importing P_{capacity} from the corresponding neighbor object. The lost electricity may then be estimated for any imported average electric power magnitude.²

Losses should be applied to only imported (i.e., purchased) electricity. The importer must account for electricity losses and the cost impacts of energy losses.³ The average power to be imported from the neighboring transactive agent must be decremented by the anticipated electricity power loss. The effective local price of the imported electric power must be increased because more power must be purchased than will, in fact, be received.

Transactions between neighboring transactive agents may also address demand charges if such practices exist between them. Neighbor class properties have been provided to configure and apply such demand charges. Specific practices may differ, but the typical practice is to add substantial monthly charges based on the highest demand that one agent is supplied by another during the prior calendar month. The demand charges attributable to relatively few high-demand periods in the month often constitute a substantial fraction of the electricity customer's monthly bill. The base transactive network template neighbor object monitors the actual demand power and compares it with highest demand power thus far in the calendar month. The demand-charge price is added to the marginal price at and above the power demand that would exceed the month's prior peak demand.⁴

¹ Marginal price has a formal definition in wholesale electricity markets, but the definition should be relaxed somewhat for use with decentralized transactive networks. Here, we simply refer to the marginal price function that results upon differentiation of utility cost functions. Marginal prices differ by location and time. The important point is that transactive network template prices should be derived everywhere using principles of cost minimization and should not include arbitrary factors or offsets that are not founded in market principles.

² The first version of the transactive network template does not account for reactive power flow, so losses may only be *estimated* while presuming constant power factor.

³ Having the agent that imports electricity incur losses and the costs of lost energy may be politically offensive, but it is computationally much more straightforward and advantageous. An agent's electricity price exists for electricity at that agent's location. Losses are a cost of moving the electricity. If the exporter were to account for losses, it would need to keep track of not only its locational price, but also all the shadow prices that would accompany the values of electricity that might be exported to many neighboring agents. If the electricity importer accounts for energy losses, it must simply compare and choose from among the values of locally generated power resources and the corrected (increased) price of importing a neighbor's electricity. A buyer naturally should make such resource decisions.

⁴ This mix of energy and demand prices is not mathematically vigorous. The intention is to forecast and approximate the real cost impacts of the demand charges in real time so that the impacts might be avoided. Some approximation is unavoidable.

2.3 What It Means to *Configure* a Transactive Network Template Code Implementation

When an implementer configures a transactive network template implementation, he establishes an agent's perspective and initializes the behaviors and properties of the market, neighbor and local asset objects with which the agent must interact. This process might eventually become somewhat automated, but the first implementations have had few, relatively time-invariant neighbor and asset objects, so automating their registry has not been needed. Instead, these objects can be instantiated and configured once using a script.

If an existing class adequately represents a needed object and its capabilities, the implementer may simply instantiate that class with the new object's name and configure its properties. If new properties or methods are needed to represent a needed object and its capabilities, an existing class must be *specialized*.

2.4 What It Means to *Specialize* a Class

A class may be specialized by creating a new class that inherits properties and methods from another. All of the inherited properties and methods may be used by an object of the new, specialized class. However, the parent class's inherited properties and methods may be redefined to suit the needs and capabilities of the new class. Specialization is to be used by the transactive network template to extend its usage to new and unique objects. For example, the transactive network template base local asset class may be specialized to represent battery energy storage. The new battery energy storage class may then be further specialized to include the utility value of battery lifetime during the scheduling process, and so on.

Specialization will be most used and useful for local asset objects. Ideally, libraries of specialized asset classes will become developed over time and made available to implementers.

Specialization should never alter the interfaces between transactive network template base classes because doing so may affect the stability of other existing implementations that use the prior transactive network template interfaces.

3.0 Suggested Future Work

The transactive network template was designed as a template for implementing transactive energy implementations in the electricity domain. Its object-oriented design establishes a structure of object types and object behaviors that will be needed to represent any circuit region within a transactive network. A useful separation of computational responsibilities has been designed into the transactive network template's base classes. Local asset objects schedule electricity consumption and generation and can thereby represent the price flexibility of an extremely diverse set of devices and systems. Market objects ensure that a local price is discovered that will balance the supply and demand of all electricity to be generated, consumed, or exchanged in the agent's circuit region. Neighbor objects manage the coordination that must occur between neighboring agents. The transactive network template further supports extensibility to address new grid objectives and, especially, to engage new local assets devices and systems and their unique scheduling needs and strategies.

The remainder of this section shall address future improvements of the transactive network template.

3.1 Power Flow Improvements

The current transactive network template version facilitates a simple pooled market approach to represent electricity exchange between transactive neighbors. Neighbors are presumed to be able to exchange electrical power within stated capacity limits. Reactive electric power and voltage are neither tracked nor managed in the transactive network. This simplification may be necessary and justified given the current reluctance and limited abilities of real-world entities to meter and, worse yet, accurately forecast their voltage or reactive power. Transport constraints and losses can be estimated as functions of electric power even though transport constraints and losses should be more accurately stated as functions of electric current. Admittedly, however, applications in microgrids and in circuits having high penetrations of intermittent renewable resources may necessitate voltage management that cannot be adequately addressed by the current transactive network template version.

Two improved versions of the transactive network template market base class are foreseen to better facilitate reactive power and voltage management. First, DC power flow principles should be applied to introduce reactive power flow and voltage and to estimate their interdependency. Voltages differences are estimated using DC power flow, so voltage constraints may be introduced and addressed upon its implementation. Reactive and real power flows are frequently decoupled in transmission system studies, where per-unit voltages lie close to unity and where transmission impedances are predominantly reactive, but this decoupling is not justified in general. The transactive network template can be applied to distribution and even smaller circuit regions. DC power flow models are generally stable, giving one hope that the decentralized application of these principles to the transactive network template might also be reliable and stable.

Eventually, a transactive network template market version using full, accurate AC power flow principles should be developed to accurately address circuit voltages. Unfortunately, the resulting equations are difficult to solve and may be found to introduce instabilities and reduced reliability to the transactive network template markets' balancing process.

The success of these improvements in the field may be limited by implementers' willingness to monitor and forecast their reactive power and voltage. Forecasts throughout the transactive network may, in fact, become less accurate if transactive agents misstate their voltages and reactive power needs and pay no penalty for doing so.

Whereas the current transactive network template version has its transactive neighbor and local assets similarly participate in the market object's balancing objective, future versions implementing DC and AC power flow principles will require new balancing calculations be used for the transactive neighbor objects. In the new versions, power exchange should not be independently asserted. Instead, power exchange is necessarily dependent upon the local circuit region's voltage, the neighbor circuit region's voltage, and the impedances of the interceding transport elements. A transactive agent may take actions to change its own complex voltage, but it cannot change other remote circuit regions' voltages in what might be a highly meshed electrical network. And its ability to do so might be heavily constrained by a constraint on local voltage, other neighbors' voltages, and constrained power flow capacities.

3.2 Facilitate Different Alternative Price-Discovery Mechanisms

The transactive network template was intended to facilitate multiple alternative price-discovery mechanisms. Its initial and current design is based on an event-based coordination of transactive signals wherein agents share complete information about their scheduled powers and residual flexibilities. It was thought that this initial formulation would be the best starting point if the transactive network template is to facilitate alternative price-discovery mechanisms. The transactive network template's ability to facilitate various price-discovery mechanisms resides within the balancing process of its market class, but it necessarily affects the timing (and perhaps content) of transactive signals, and it may also affect the timing and information being exchanged with local assets as they are scheduled.

A fortuitous design decision may limit the impacts of changing price-discovery mechanisms on local asset objects. Having asserted that local asset objects' properties and strategies are fully known by their transactive agent, we might further assert that changing the price-discovery mechanism need not significantly change the scheduling of local asset objects. An implementer might simply be required to make all local assets return their residual flexibility as described in this report. Unfortunately, future implementers might (will) refuse to do so.

If any single local asset (or transactive neighbor, for that matter) cannot (or chooses to not) calculate its residual flexibility, then a more general method using sub-gradient search and iteration must be employed by the balancing process. This alternative balancing method has already been coded into the base transactive network template market class. In the author's opinion, this alternative method is unnecessary, computationally iterative, and should be avoided. A local asset having hidden properties and unknown scheduling strategy should be recast as an independent transactive neighbor agent.

A change in price-discovery mechanism will be outwardly visible in the contents and timing of transactive signals between transactive neighbors. The author is hopeful that most auctions (examples of alternative price-discovery mechanisms) can be implemented by further constraining the timing and signal content of transactive signals while requiring few changes to the base market class's balancing process. Price-discovery mechanisms derived from game theory (e.g., consensus methods) will require more effort and may require more changes to the balancing and even scheduling processes.

Transactive signals are more important to the signal's recipient than its sender. The timing and contents of sent signals should meet the needs of the intended recipient's price-discovery process and mechanism. In general, the same price-discovery mechanism should be employed throughout a transactive network. If this were not the case, the network's transactive agents could not be sure that their objectives will be incentivized, negotiated, and conveyed through the network. The neighboring agents' forecast horizons

might differ. One agent might be invoked to iterate the exchange of transactive signals when such iterations are not meaningful there and are otherwise unnecessary for its purposes.

3.3 Support of Sequential Correction Markets

The transactive network template balancing process represents a distributed, decentralized market process. A single market is certainly interesting, but a need will likely arise to employ multiple sequential markets, where successive markets refine and correct the outcomes from the earlier ones. This situation is common among wholesale electricity markets, where long-term markets become successively corrected by day-ahead, intra-day, hourly, intra-hour, real-time, and reconciliation markets. The current market object design was given properties that anticipate this need, but the capability has not been tested or implemented yet.

3.4 Support of Financial Transactions

Pilot implementations of transactive systems have had their dynamic locational prices enforced to differing degrees. The transactive networks' dynamic prices have frequently been permitted to diverge from electricity billing practices and are therefore ignored or must be corrected when calculating actual customer bills. The current transactive network template design anticipated flags to mark agents' commitments to prices and quantities. A reconciliation market state was created to allow time for market outcomes to become resolved and settled. The revenue implications and practices for those who implement and participate in transactive networks may be significantly more dynamic than those that predominate today. Transition to a transactive world will be considered risky. The connections between transactive network processes and electricity billing practices must be facilitated and tested.



**Pacific
Northwest**
NATIONAL LABORATORY

www.pnnl.gov

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99352
1-888-375-PNNL (7665)

U.S. DEPARTMENT OF
ENERGY