Introductory Notes on Agent-Based Modeling, Agent-Oriented Programming, & AMES

Presenter:

Leigh Tesfatsion

Professor of Economics

Courtesy Professor of Mathematics and

Electrical & Computer Engineering

Iowa State University

Ames, Iowa 50011-1070

http://www.econ.iastate.edu/tesfatsi/

tesfatsi@iastate.edu

Last Updated: 13 June 2010

https://www2.econ.iastate.edu/tesfatsi/ABMAOPAMES.LT.pdf

Outline

- What is Agent-Based Modeling (ABM)?
- What is Object-Oriented Programming (OOP)?
- Agent-Oriented Programming (AOP) vs. OOP
- ABM via Computational Laboratories
- **Example:**
  AMES Wholesale Power Market Testbed

What is ABM?

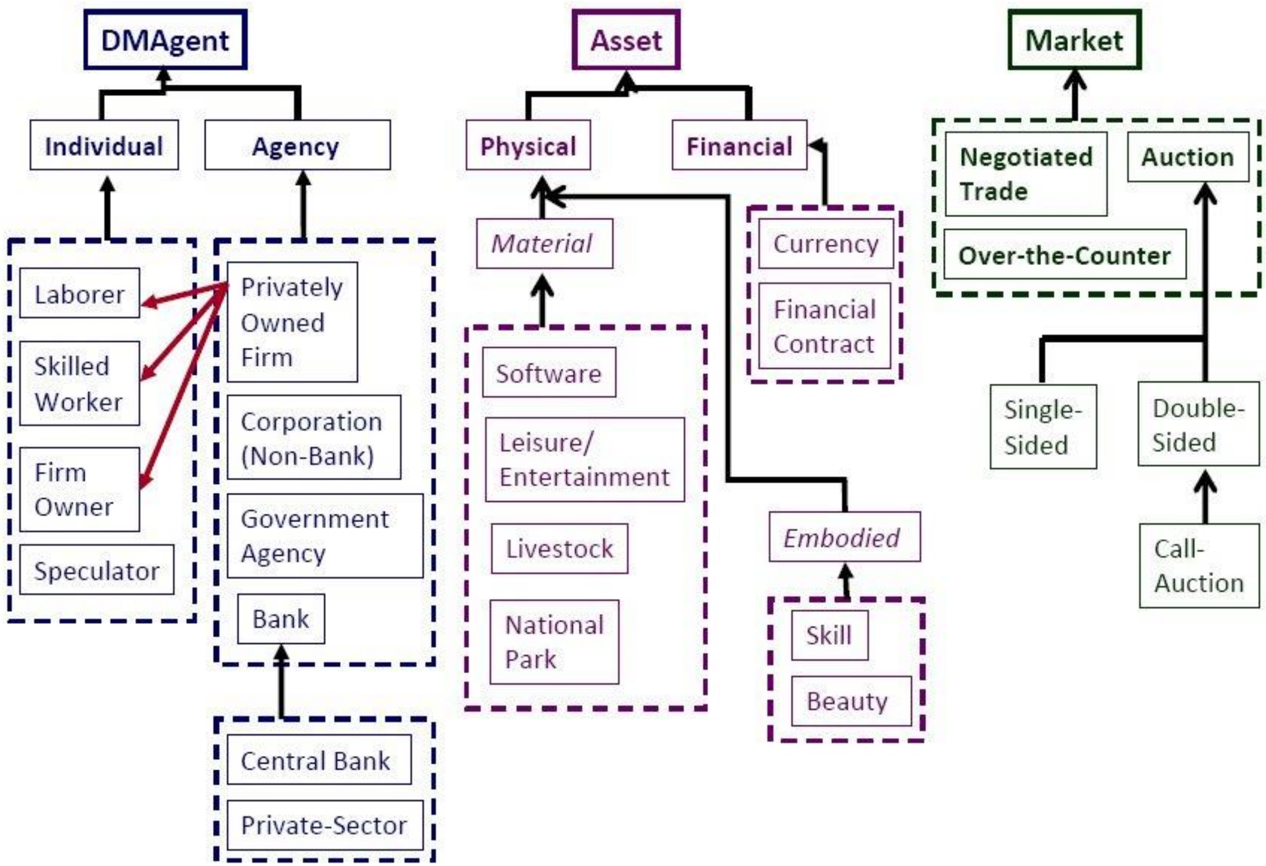
- Classical Approach (Top Down): Model a system by means of parameterized differential equations
  - *Example: Archimedes*, a large-scale system of ODEs modeling pathways of disease spread under alternative possible health care response systems
- ABM Approach (Bottom Up): Model a system as a collection of interacting "agents"
  - Each agent is an entity encapsulating data together with methods that act on this data
  - Global regularities arise from the interactions of distributed agents

Meaning of “Agent” in ABM

Agent = Encapsulated bundle of data and methods acting within a computationally constructed world.

- Agents can represent:
  - Individuals (consumers, traders, entrepreneurs,…)
  - Social groupings (households, communities,…)
  - Institutions (markets, corporations, gov't agencies,…)
  - Biological entities (crops, livestock, forests,…)
  - Physical entities (weather, landscape, electric grids,…)

Partial depiction of agents for a macroeconomic ABM with "is a" ↑ and "has a" ↓ relations



Meaning of "Agent" in ABM … Continued

Decision-making agents (DMAgents) are capable (in different degrees) of

- Behavioral adaptation
- Goal-directed learning
- Social communication (talking with each other!)
- Endogenous formation of interaction networks
- **Autonomy:**

  Self-activation and self-determination based on ***private internal*** data and methods as well as on external data streams (including from real world)

Importance of Agent Encapsulation

- In the real world, all calculations must be done by entities actually residing in the world.
- ABM forces modelers to respect this constraint.
- Procedures encapsulated into the methods of a particular agent can only be implemented using the particular resources available to that agent.
- This encapsulation achieves a more transparent and realistic representation of real-world systems composed of interacting distributed entities with limited information and computational capabilities.

Constructive Replacement

In principle, as a result of agent encapsulation:

- **Any decision-making agent** interacting within an ABM through a particular input-output public interface can be replaced by a person that interacts with the ABM through this same public interface.

- Since method *implementations* by the decision-making agent and its human replacement need not be the same, the resulting outcomes under replacement could differ.

- The only claim here is the *feasibility* of replacement due to the imposition of agent boundaries in ABMs.

Role of Equations

- Any agent in an ABM can have data and/or methods involving equations.
- These equations can be the basis in part or in whole for the agent's actions.
- ABM world events are driven solely by the **actions** undertaken by the ABM agents within their world.
- ABM world events are ***not*** driven by equations existing ***outside*** of the data and methods of agents. For example, "sky hook" equilibrium conditions are not permitted.

ABM and Institutional Design

Key Issues:

- Will a proposed or actual design promote **efficient, fair, and orderly social outcomes over time**?
- Will the design give rise to **unintended consequences**?

ABM Culture-Dish Approach:

- **Develop a computational world** embodying the design, physical constraints, strategic participants, …
- Set initial world conditions (agent states).
- **Let the world evolve** with no further intervention, and observe and evaluate the resulting outcomes.

Agent-Based Test Bed Development via Iterative Participatory Modeling

- Stakeholders and researchers from multiple disciplines join together in a **repeated looping through four stages of analysis**:

1) Field work and data collection
2) Role-playing games/human-subject experiments
3) Incorporate findings into agent-based test bed
4) Generate hypotheses through intensive computational experiments.

Object-Oriented Programming

The Basic OOP Mindset: Top Down

"One of the best ways to think about objects is as `service providers.' Your goal is to produce…a set of objects that provides the ideal services to solve your problem."

– Bruce Eckel, *Thinking in Java*, 3rd Ed., Prentice Hall, 2003, p. 37.

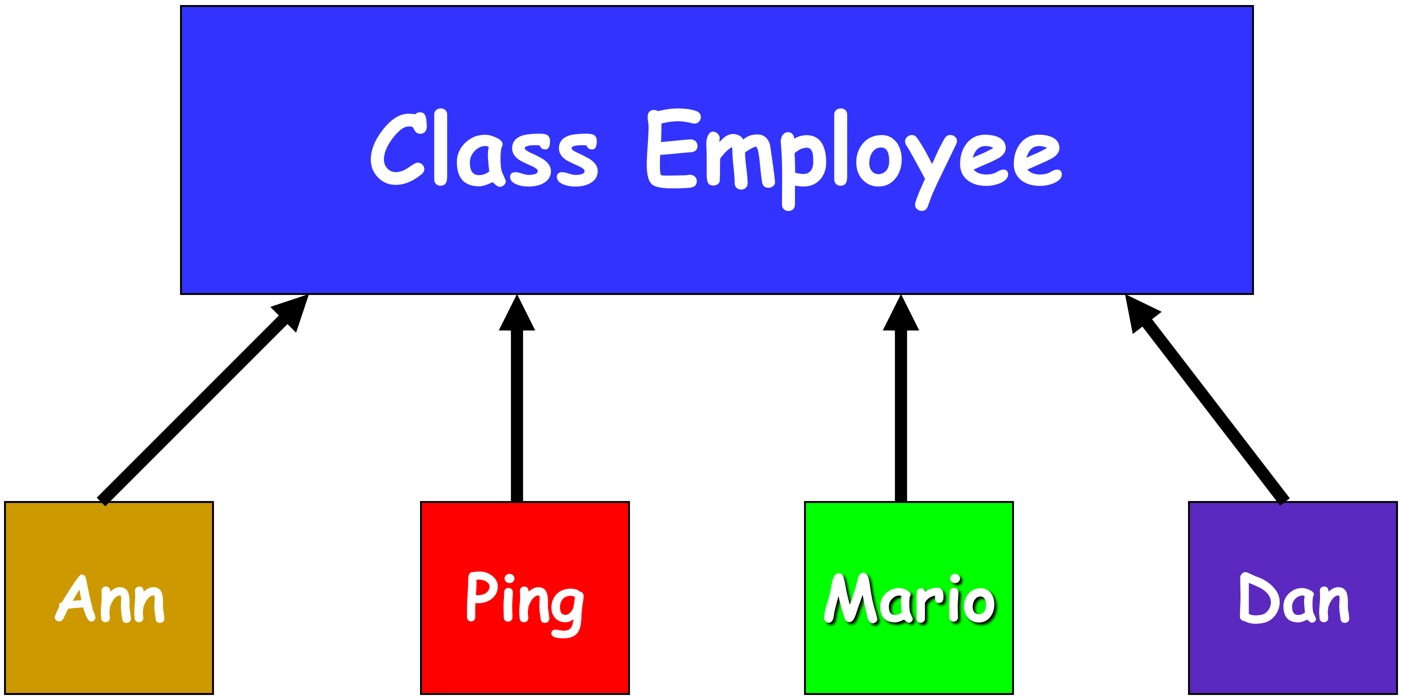
OOP: Key Concepts

- **Object**
  - Methods (operations, functions, procedures,...)
  - Data (attributes, state information,...)
  - Access (public, private, or protected)
- **Class**
- **Interface**
- **Encapsulation**
- **Inheritance (subclass, superclass)**
- **Composition**

OOP: Key Concepts … Continued

- An ***object*** is a software entity that bundles together ***data*** plus ***methods*** that act on these data
- An object controls ***access*** to its data and methods by declaring them
  - public (accessible to all other objects);
  - private (inaccessible to all other objects);
  - or protected (accessible only to certain designated other objects).
- A ***class*** is a blueprint for an object, a template used to create ("instantiate") an object.

Class = Object Template



Employee Objects (Instances of Employee)

Illustration: Employee Class

Class **EMPLOYEE**

{

Public Access:

Methods:

getSocialSecurityNumber( ) ;

getGender( ) ;

getDateOfBirth( ) ;

Private Access:

Data:

SocialSecurityNumber ;

Gender ;

DateOfBirth ;

Trustworthyness ;

}

OOP: Key Concepts … Continued

- The public methods and public data of an object are called its ***(public) interface***.
- Objects ***communicate*** with each other by activating ("invoking") the public methods of other objects.

OOP: Key Concepts … Continued

- In "good" OOP design, an object should only reveal to other objects what these objects need to know to interact with it.
- Each class specifies interfaces for its instantiated objects
  - Describes how users of these instantiated objects can interact with these instantiated objects.

Illustration: Employee Class

```
Class EMPLOYEE
{
 Public Access:
   Methods:
      getSocialSecurityNumber( ) ;
      getGender( ) ;
      getDateOfBirth( ) ;
 Private Access:
   Data:
      SocialSecurityNumber ;
      Gender ;
      DateOfBirth ;
      Trustworthyness ;
}
```

Illustration: Payroll Class
(invokes public methods in Employee class)

```
Class  PAYROLL
{
 Public Access:
  Methods:
    calculateEmployeePay( );
    payEmployee( );
    Employee.getSocialSecurityNumber( );
    Employee.getGender( );
    Employee.getDateOfBirth( );

Private Access:
 Data:
   CurrentProfits;
   EmployeePayoll;
}
```

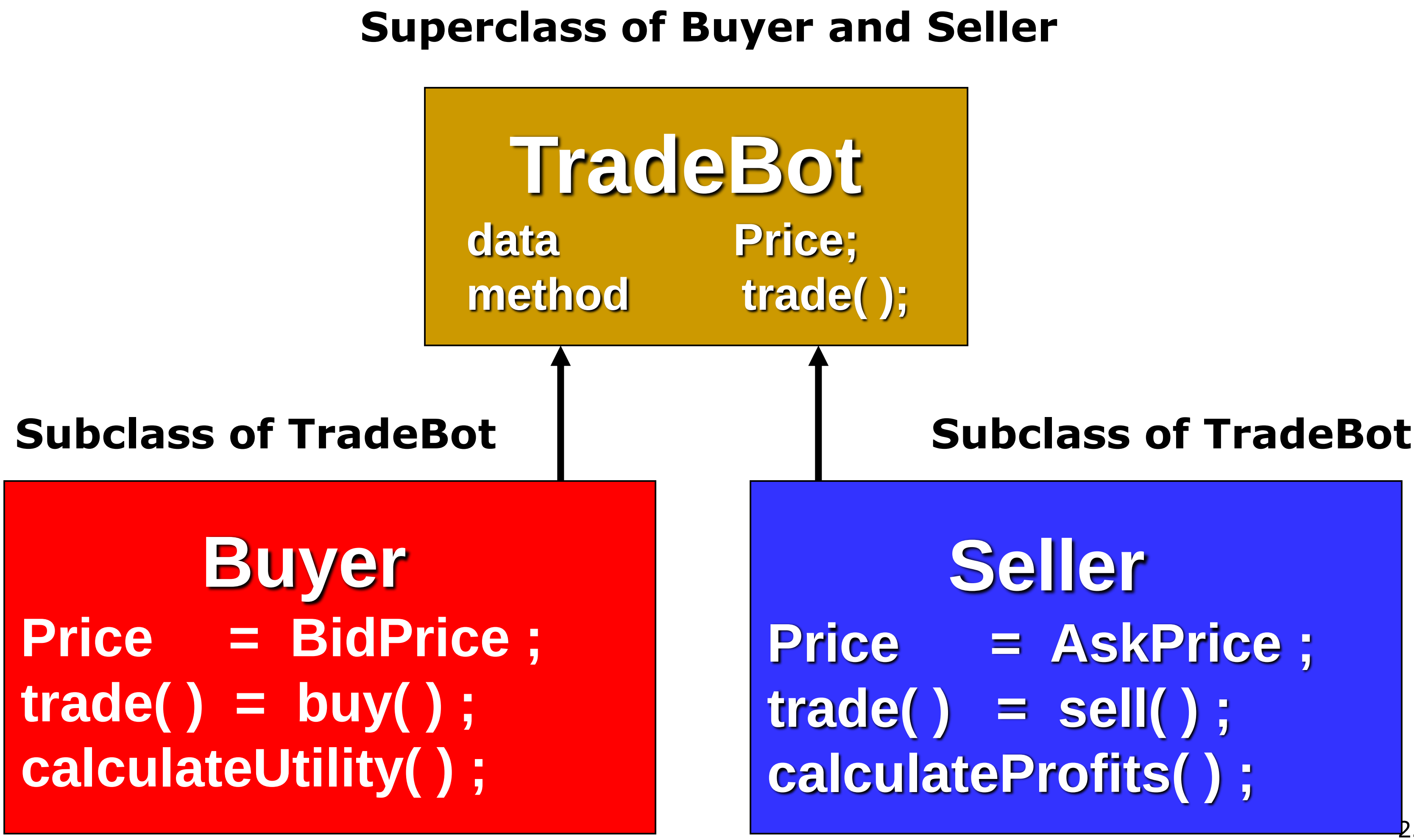
OOP Encapsulation

- *Encapsulation* is the process of determining which aspects of a class are not needed by other classes, and hiding these aspects from other classes.
- More precisely, encapsulation is the process of dividing a class into two distinct parts:
  - (public) interface;
  - private (or protected) stuff that other classes do not need to know about.

Class Inheritance

- A class C can ***inherit*** the data and methods of another class B.
- The class C is then called the ***subclass*** of class B, and class B is called the ***superclass*** of class C.
- A subclass can also include specialized data and methods that are not present in the superclass.

Class Inheritance: Example



Composition vs. Inheritance

- Objects can be built, or "composed", from other objects. This is called ***composition***.

  Example: A firm is composed of employees.

- A ***composition*** relationship between objects is often termed a ***"Has-A" relationship***. A firm "has an" employee.
- An ***inheritance relationship*** between objects is often termed an ***"Is-A" relationship***. A buyer "is a" trader.

Agent-Oriented Programming (AOP) vs. Conventional OOP

- What is an ***agent*** in AOP?
- How does an AOP agent *potentially extend* conventional OOP objects?

What is an *Agent* in AOP?

AOP agent is an object *potentially* capable of ...

- **(Structural) Reactivity:** Changes in internal structure in response to environmental changes
- **Social Ability:** Interaction with other agents through some form of language.
- **Pro-Activity:** Goal-directed actions.
- **Autonomy:** Some degree of control over its own actions ("self-activation").

AOR Agents vs. OOP Objects

- Key distinction is autonomy of AOR Agents
- AOR agents can potentially exercise distributed control, not simply distributed action.
- Conventional OOP objects encapsulate data and methods but do not permit either self-activation or local action choice.

Cf. N. R. Jennings, *Artificial Intelligence*, Vol. 17 (2000), pp. 277-296

Autonomy means...

- Each agent effectively has its own persistent thread of control.
- Each agent decides which actions to perform when, based partly on external conditions ***and partly on private internal aspects*** (private info, beliefs, desires,…).
- Thus, each agent is imperfect predictable from vantage point of other agents due to ***behavioral uncertainty***.

ABM via Comp Laboratories

- ***Computational Laboratory*** = Computational framework for the study of system behaviors by means of controlled & replicable experiments.
- ***Graphical User Interface (GUI)*** permits experimentation by users with no programming background.
- ***Modular/extensible form*** permits framework capabilities to be changed/extended by users who have programming background.

Illustrative Example:

AMES Wholesale Power Market Test Bed

Project Director: **Dr. Leigh Tesfatsion** (Prof of Econ, Courtesy Prof. of Math & ECpE)

Research Assoc: **Dr. Junjie Sun** (Fin. Econ, OCC, U.S. Treasury, Wash, D.C.)

Research Assoc: **Dr. Hongyan Li** (Consulting Eng., ABB, North Carolina)

funded in part by the

National Science Foundation and the ISU

Electric Power Research Center (a power industry consortium)

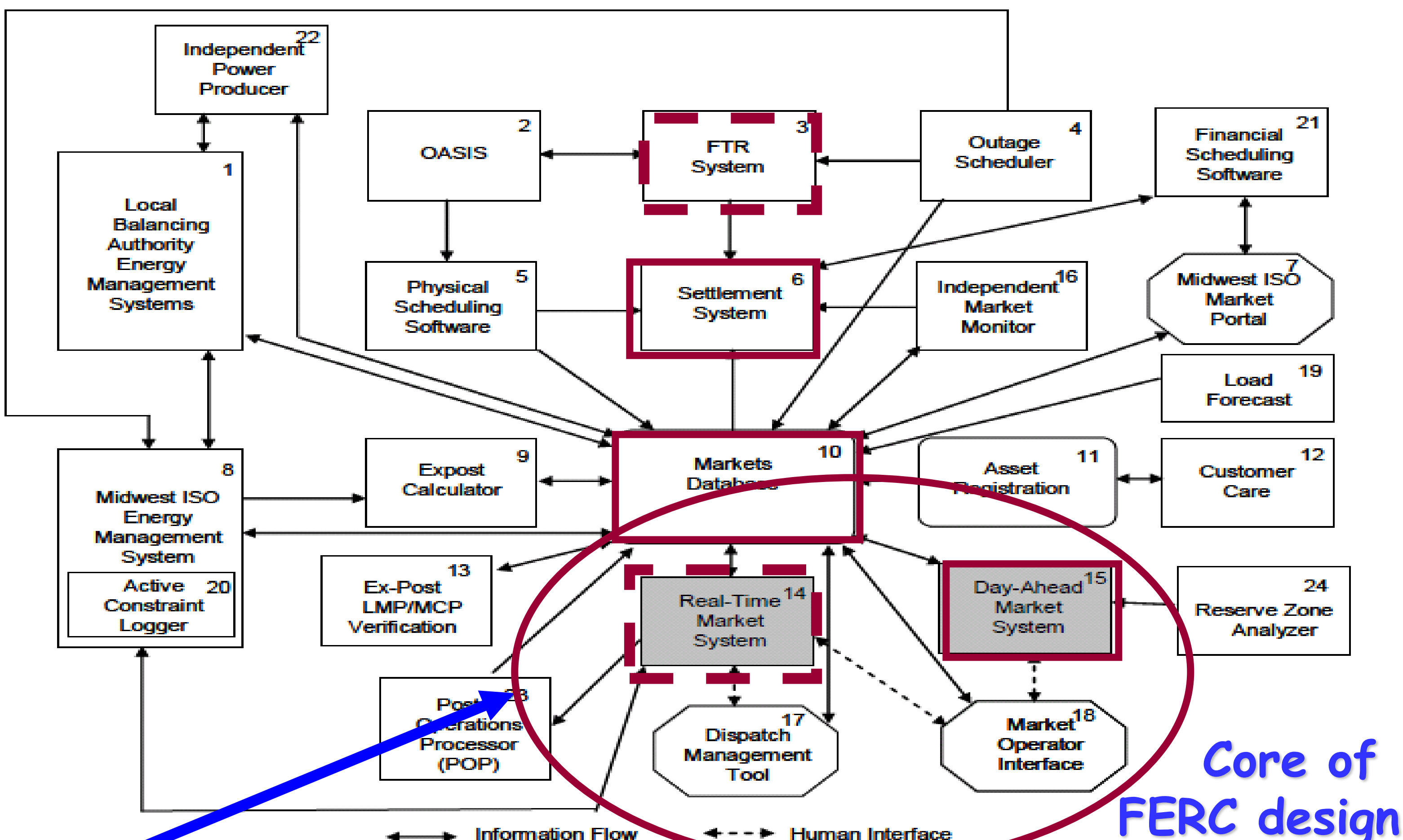
AMES Wholesale Power Market Test Bed Homepage (Code/Manuals/Pubs):
https://www2.econ.iastate.edu/tesfatsi/AMESMarketHome.htm

AMES (V2.05) Wholesale Power Market Testbed

- **Traders**
  - GenCos (sellers)
  - LSEs (buyers)
  - **Learning** capabilities
- **Independent System Operator (ISO)**
  - System reliability assessments
  - Day-ahead scheduling via **bid/offer based optimal power flow (OPF)**
  - Real-time dispatch
- **Two-settlement system**
  - Day-ahead market (double auction, financial contracts)
  - Real-time market (settlement of differences)
- **AC transmission grid**
  - **Generation Companies (GenCos)** & **Load-Serving Entities (LSEs)** located at user-specified transmission buses
  - Grid congestion managed via **Locational Marginal Prices (LMPs)**
  - **LMP at bus k** = Least cost of servicing one additional MW of power at bus k.

Based on MISO Market Organization
Business Practices Manual 001-r1 (1/6/09)

Exhibit 2-3: DART Components Overview



Two-Settlement Power Market System under LMP

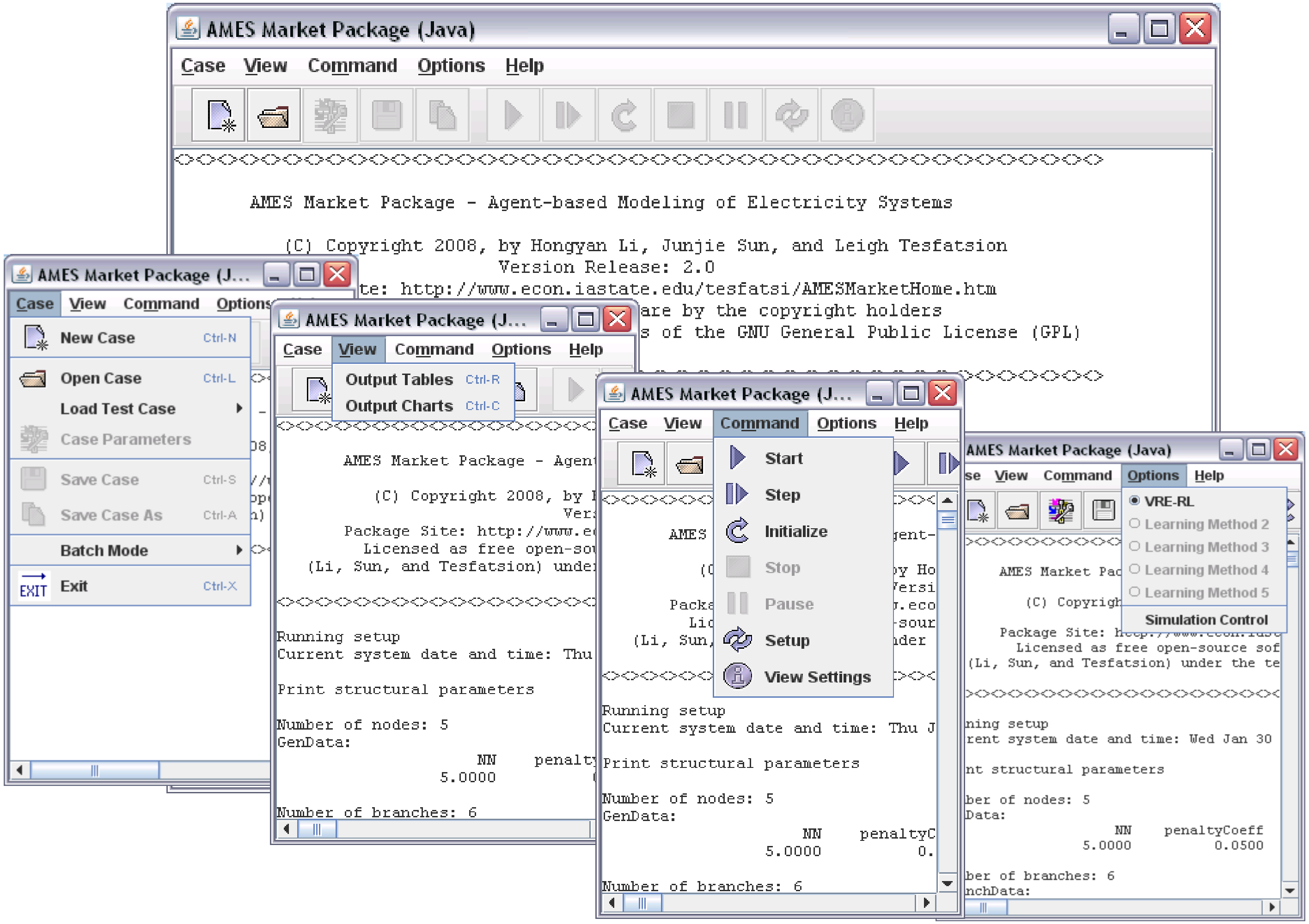
AMES project to date

AMES Modular & Extensible Architecture (Java)

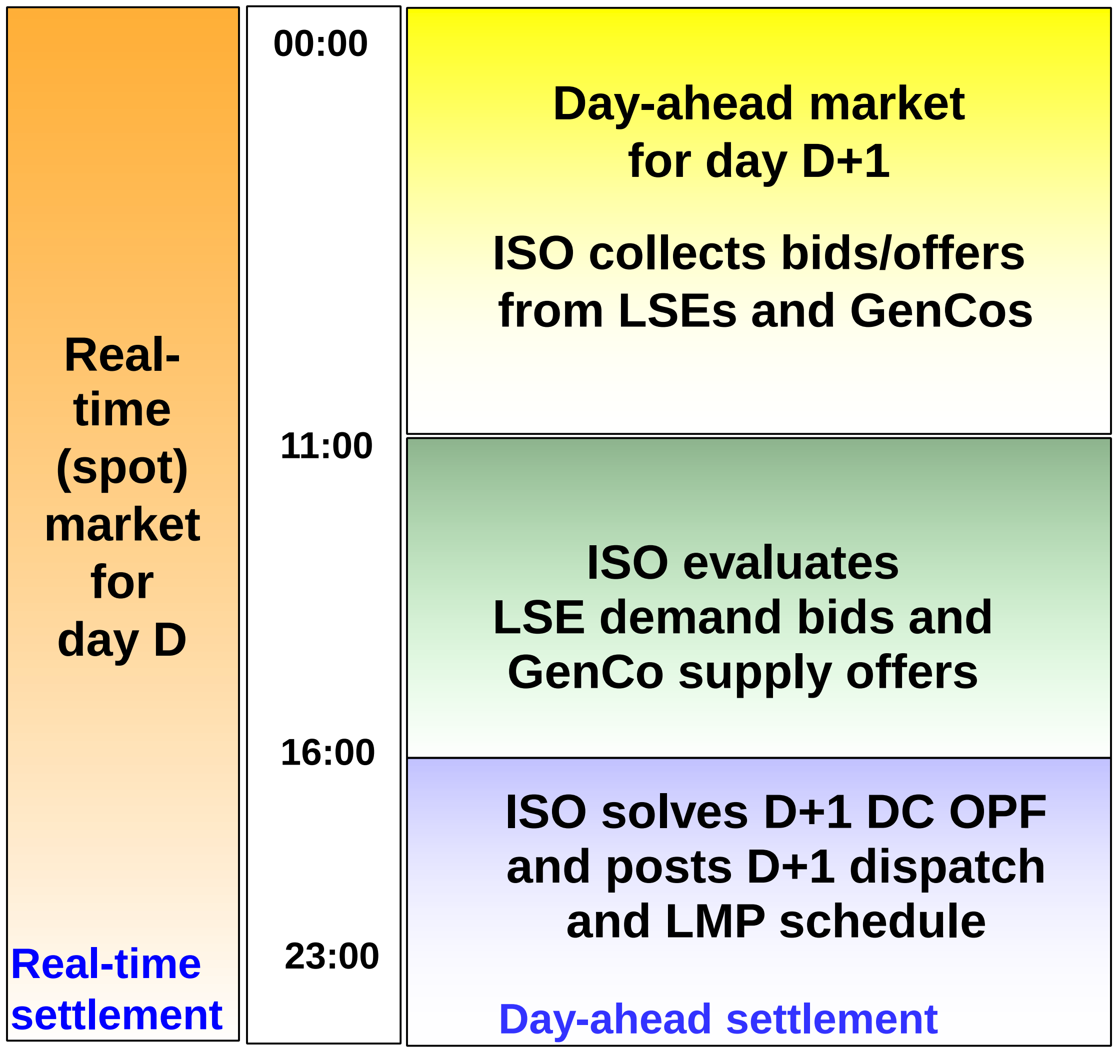
- **Market protocols & AC transmission grid structure**
  - **Graphical user interface (GUI) & modularized class structure** permit easy experimentation with alternative parameter settings and alternative institutional/grid constraints
- **Learning representations for traders**
  - **Java Reinforcement Learning Module (JReLM)**
  - "Tool box" permitting experimentation with a wide variety of learning methods (Roth-Erev, Temp Diff/Q-learning,…)
- **Bid/offer-based optimal power flow formulation**
  - **Java DC Optimal Power Flow Module (DCOPFJ)**
  - Permits experimentation with various DC OPF formulations
- **Output displays and dynamic test cases**
  - Customizable chart/table displays & 5-bus/30-bus test cases

AMES Graphical User Interface (GUI)

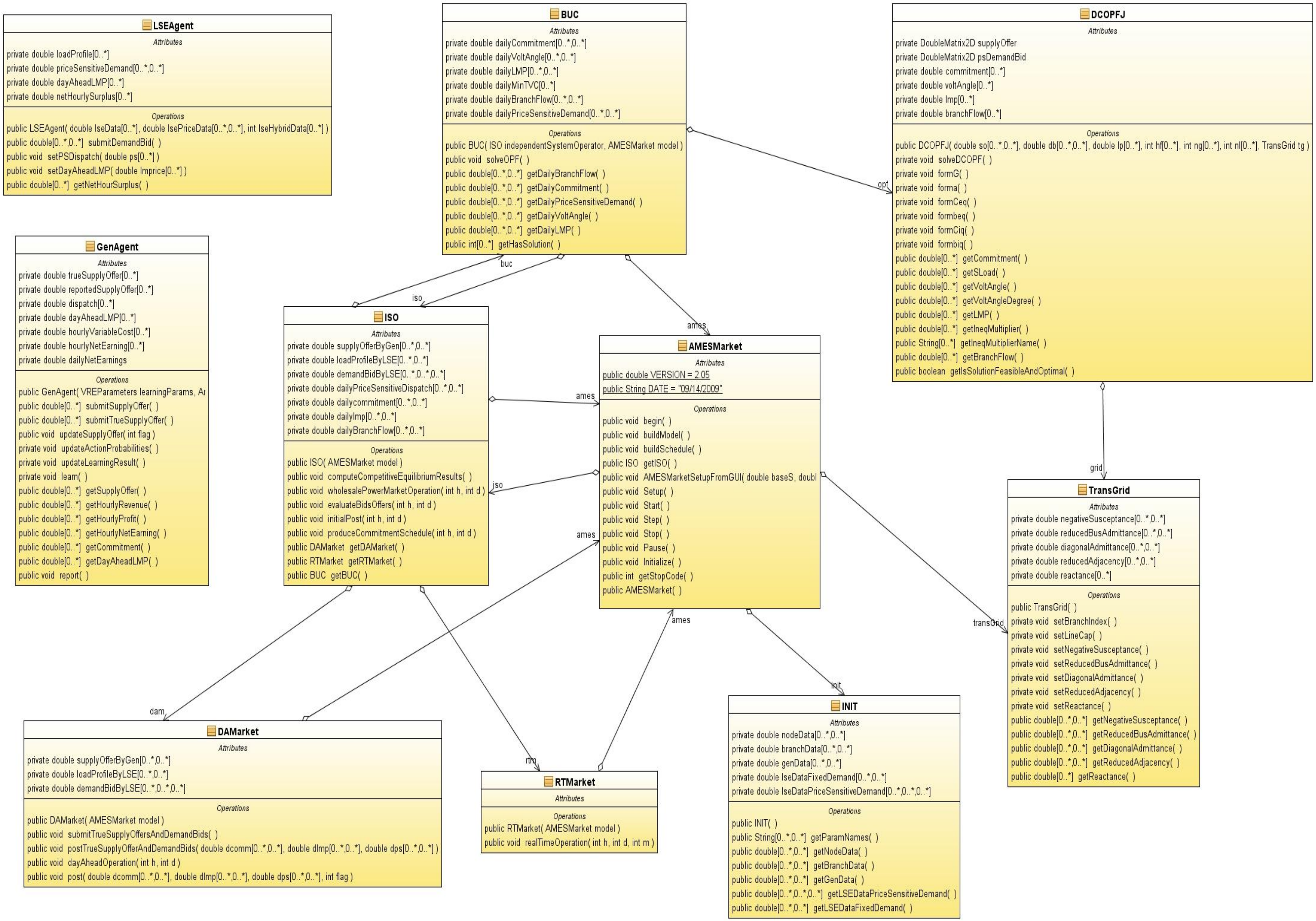
Tool Bar and Menus for Data Input and Output Displays



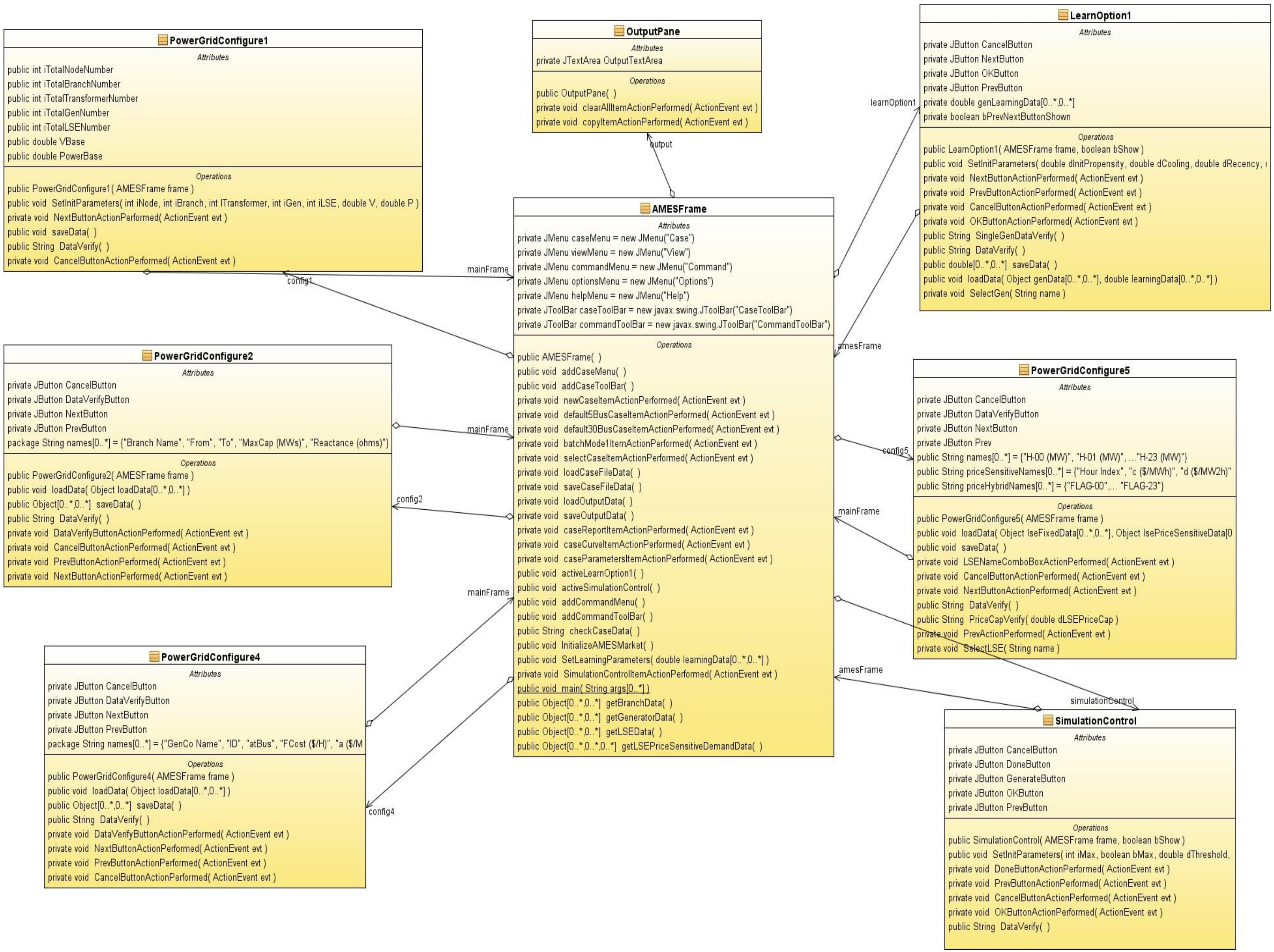
Activities of AMES ISO During Each Operating Day D: Timing Adopted from Midwest ISO (MISO)



Basic AMES Classes:



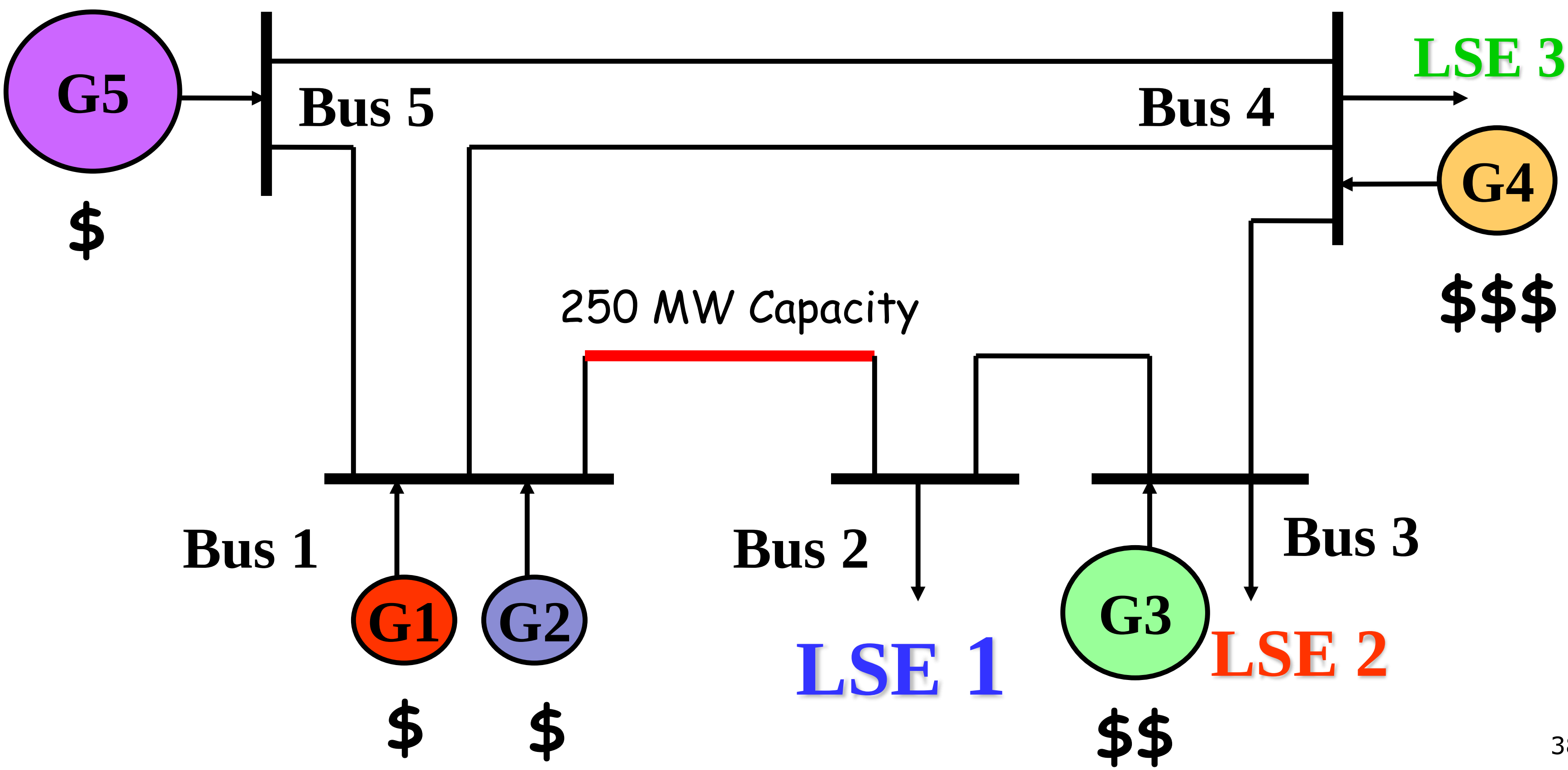
AMESFrame Classes (GUI)



Illustrative 5-Bus Test Case

(Used in many ISO business practice/training manuals)

Five GenCo sellers G1,…,G5 and three LSE buyers LSE 1, LSE 2, LSE 3



Partial depiction of input data for the 5-bus test case:

Base Values[a]	
S_o	V_o
100	10
K^b	π^c
5	0.05

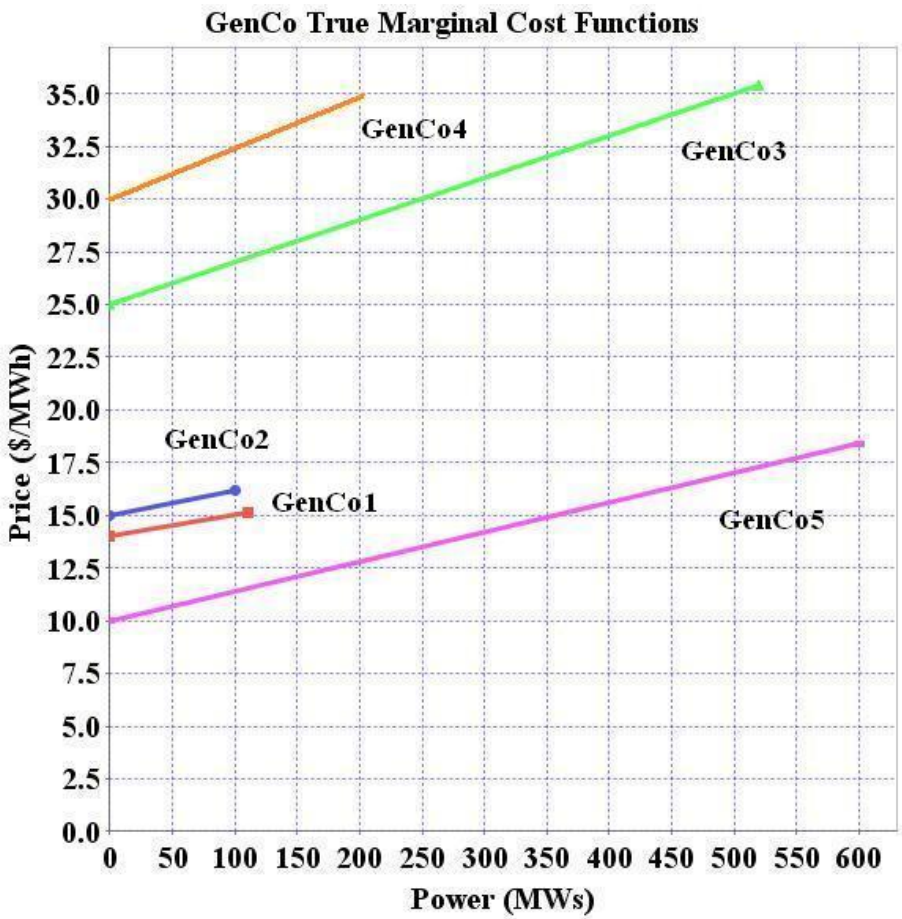
Branch From	To	lineCap[d]	X^e
1	2	250.0	0.0281
1	4	150.0	0.0304
1	5	400.0	0.0064
2	3	350.0	0.0108
3	4	240.0	0.0297
4	5	240.0	0.0297

Gen ID	atNode	FCost	a	b	Cap^L	Cap^U	Init$
1	1	1600.0	14.0	0.005	0.0	110.0	$\$1M$
2	1	1200.0	15.0	0.006	0.0	100.0	$\$1M$
3	3	8500.0	25.0	0.010	0.0	520.0	$\$1M$
4	4	1000.0	30.0	0.012	0.0	200.0	$\$1M$
5	5	5400.0	10.0	0.007	0.0	600.0	$\$1M$

LSE ID	atNode	L-00[f]	L-01	L-02	L-03	L-04	L-05	L-06	L-07
1	2	350.00	322.93	305.04	296.02	287.16	291.59	296.02	314.07
2	3	300.00	276.80	261.47	253.73	246.13	249.93	253.73	269.20
3	4	250.00	230.66	217.89	211.44	205.11	208.28	211.44	224.33
ID	atNode	L-08	L-09	L-10	L-11	L-12	L-13	L-14	L-15
1	2	358.86	394.80	403.82	408.25	403.82	394.80	390.37	390.37
2	3	307.60	338.40	346.13	349.93	346.13	338.40	334.60	334.60
3	4	256.33	282.00	288.44	291.61	288.44	282.00	278.83	278.83
ID	atNode	L-16	L-17	L-18	L-19	L-20	L-21	L-22	L-23
1	2	408.25	448.62	430.73	426.14	421.71	412.69	390.37	363.46
2	3	349.93	384.53	369.20	365.26	361.47	353.73	334.60	311.53
3	4	291.61	320.44	307.67	304.39	301.22	294.78	278.83	259.61

[a] For simplicity, the base apparent power S_o (MVA) and base voltage V_o (kV) are chosen so base impedance Z_o satisfies $Z_o = V_o^2/S_o = 1$.
[b] Total number of nodes
[c] Soft penalty weight π for voltage angle differences
[d] Upper limit P_{km}^U (in MWs) on the magnitude of real power flow in branch km
[e] Reactance X_{km} (in ohms) for branch km
[f] L-H: Load (in MWs) for hour H, where H=00,01,...,23

GenCo True Cost & Capacity Attributes for the 5-Bus Test Case



AMES Generation Company (Seller)

Public Access:

// Public Methods

getWorldEventSchedule(clock time,…);
getMarketProtocols(ISO market power mitigation,…);
Methods for receiving data;
Methods for retrieving GenCo data;

Private Access:

// Private Methods

Methods for gathering, storing, and sending data;
Methods for calculating own expected & actual net earnings;
Method for updating own supply offers (**LEARNING**);

// Private Data

Own capacity, grid location, cost function, current wealth… ;
Data recorded about external world (prices, dispatch,…);
Address book (communication links);

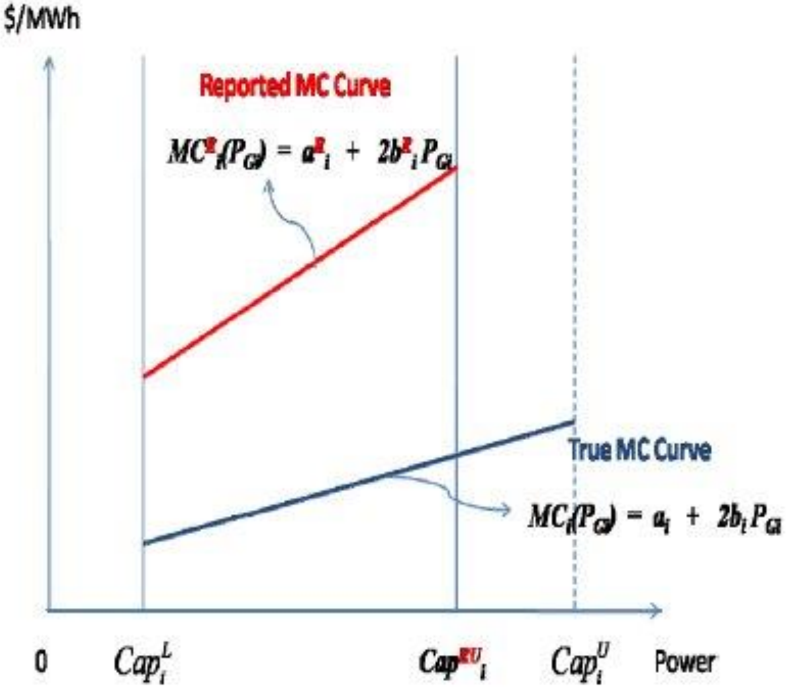
AMES GenCos are learners who report strategic hourly supply offers to ISO

Hourly supply offer for each GenCo i = ***Reported*** linear marginal cost function over a ***reported*** operating capacity interval for real power p_{Gi} (in MWs):

$$MC^{R}_{i}(p_{Gi}) = a^{R}_{i} + 2b^{R}_{i}\, p_{Gi}$$

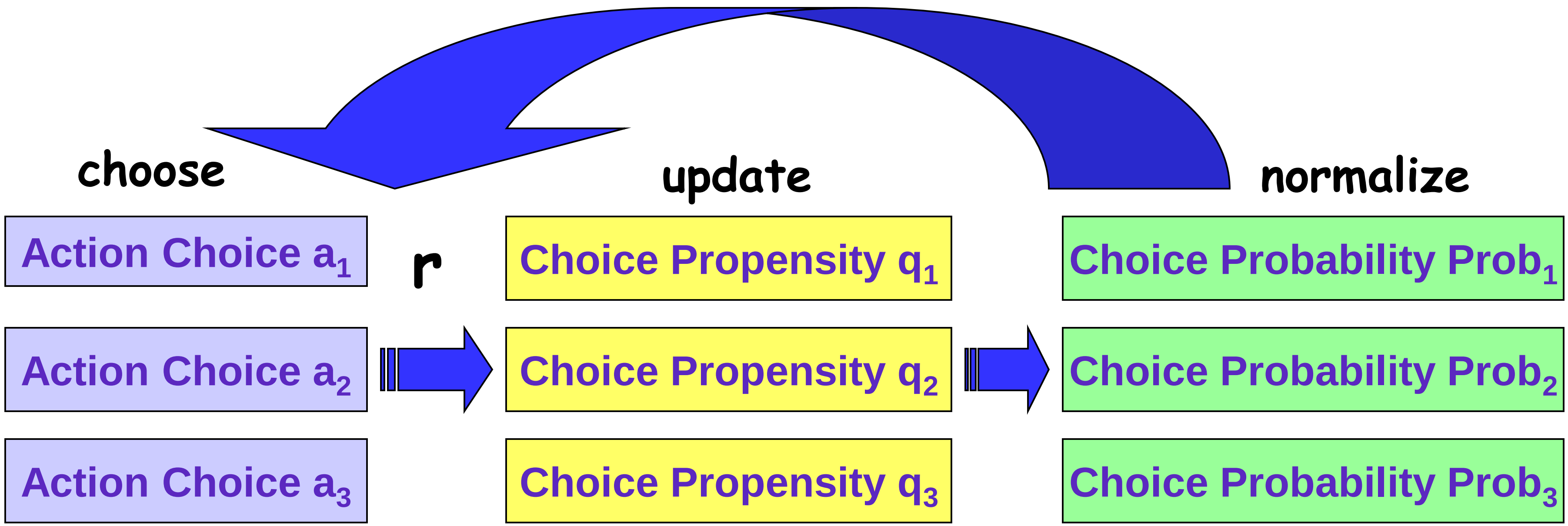
$$Cap_{i}^{L} \leq p_{Gi} \leq Cap_{i}^{RU}$$

GenCos can learn to report ***higher-than-true*** marginal costs and/or to report ***lower-than-true*** maximum capacity.



In 5-bus study, AMES GenCos use VRE learning

(version of Roth-Erev stochastic reinforcement learning)



- Each GenCo maintains action choice propensities q, normalized to choice probabilities Prob, to choose actions (supply offers). *A* good (bad) reward r_k resulting from an action a_k results in an increase (decrease) in both q_k and $Prob_k$.

VRE Updating of Action Propensities

Parameters:

- $q_j(1)$ Initial propensity
- ε Experimentation
- φ Recency (forgetting)

Variables:

- a_j Current action choice
- q_j Propensity for action a_j
- a_k Last action chosen
- r_k Reward for action a_k
- t Current time step
- N Number of actions

$$q_j(t+1) = [1-\phi]q_j(t) + E_j(\varepsilon,N,k,t)$$

$$E_j(\varepsilon,N,k,t) = \begin{cases} r_k(t)[1-\epsilon] & \text{if } j = k \\ q_j(t)\frac{\epsilon}{N-1} & \text{if } j \neq k \end{cases}$$

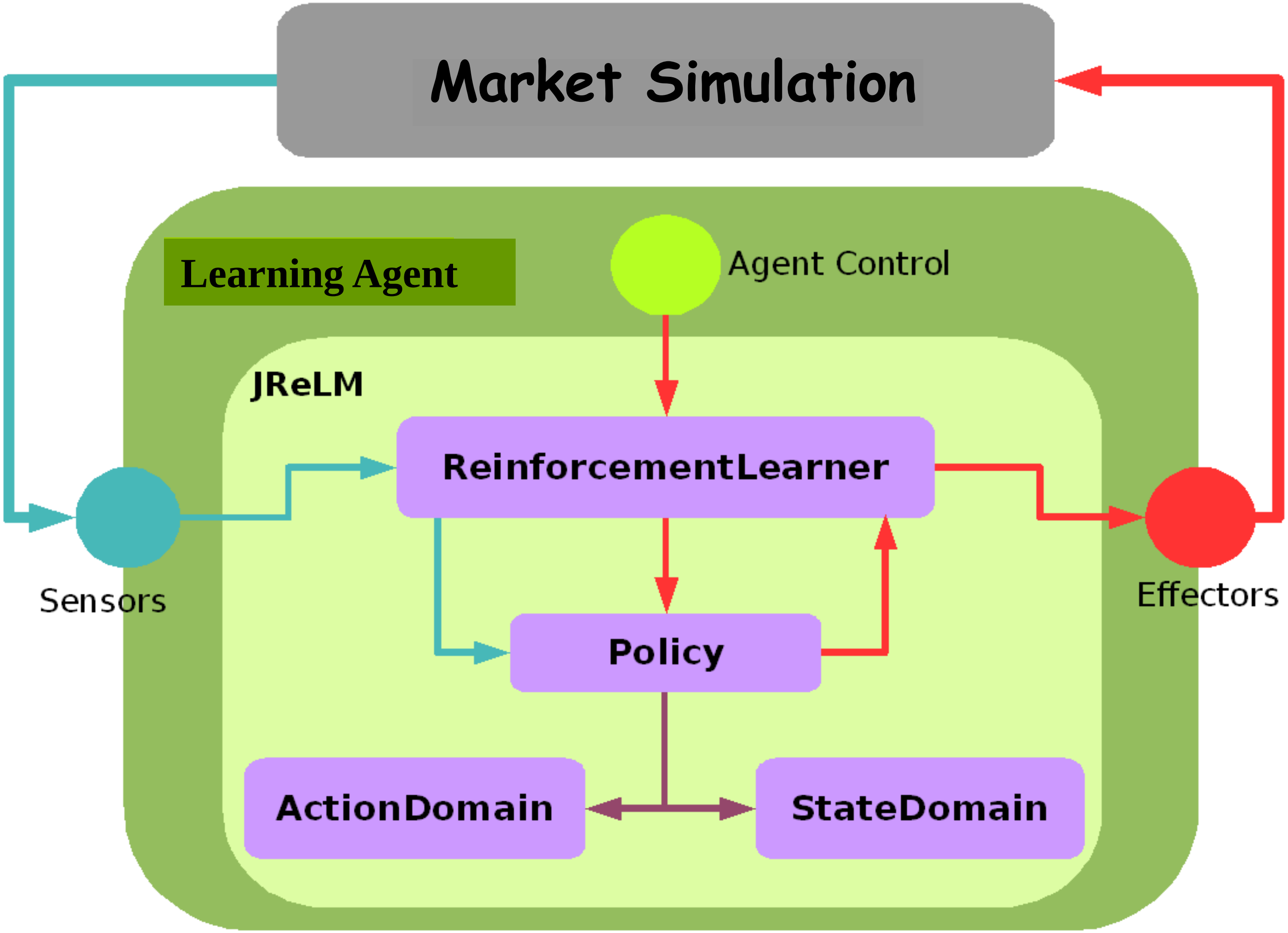
From VRE Propensities to Probabilities
Gibbs-Boltzmann Probability

- $q_j(t)$ = "propensity" to choose action j at time t
- N = Total number of available actions
- T = Temperature ("cooling") parameter
- $Prob_j(t)$ = Probability of choosing action j at time t

$$Prob_j(t) = \frac{e^{q_j(t)/T}}{\sum_{n=1}^{N} e^{q_n(t)/T}}$$

AMES GenCo learning implemented via JReLM module

(Java Reinforcement Learning Module developed by Charles J. Gieseler, Comp Sci M.S. Thesis, 2005)



AMES Load-Serving Entity (Buyer)

Public Access:

// **Public Methods**

getMarketProtocols(posting, trade, settlement);
getMarketProtocols(ISO market power mitigation);
Methods for receiving data;
Methods for retrieving LSE data;

Private Access:

// **Private Methods**

Methods for gathering, storing, and sending data;
Methods for calculating own expected & actual net earnings;

// **Private Data**

Own downstream demand, grid location, current wealth…;
Data recorded about external world (prices, dispatch,…);
Address book (communication links);

AMES LSE Hourly Demand-Bid Formulation

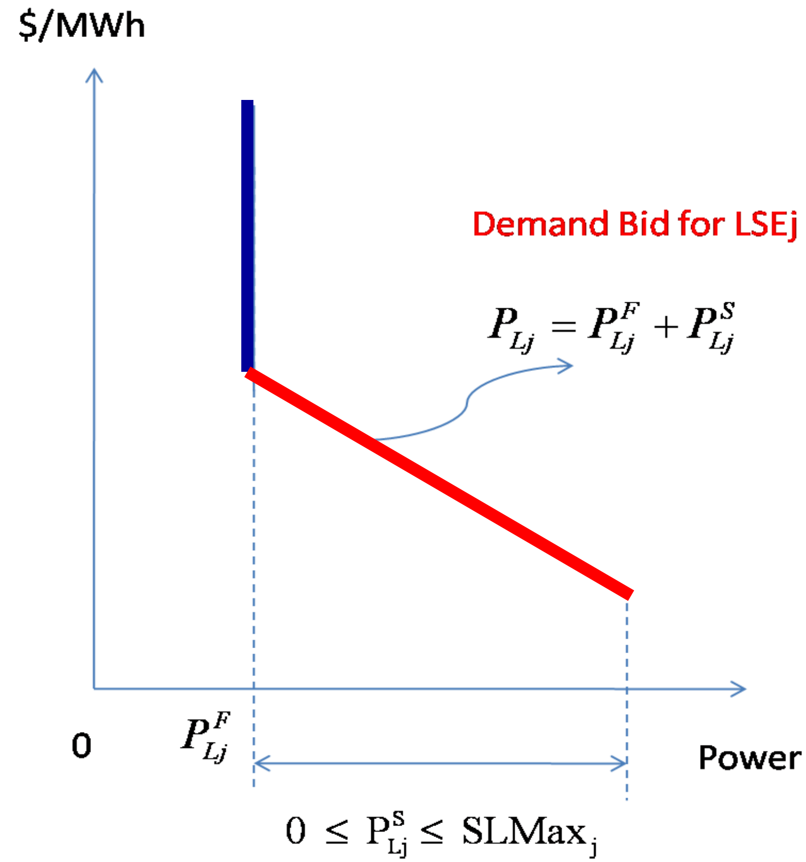
- Hourly demand bid for each LSE j

Fixed* + *Price-Sensitive Demand Bid

- **Fixed** demand bid = p^F_{Lj} (MWs)
- **Price-sensitive** demand bid = Inverse demand function for real power p^S_{Lj} (MWs) over a purchase capacity interval:

$$F_j(p^S_{Lj}) = c_j - 2d_j\, p^S_{Lj}$$

$$0 \le p^S_{Lj} \le SLMax_j$$



AMES ISO (Market Operator)

Public Access:

// Public Methods

getWorldEventSchedule(clock time,…);
getMarketProtocols(bid/offer reporting, settlement,…);
Methods for receiving data;
Methods for retrieving stored ISO data;

Private Access:

// Private Methods

Methods for gathering, storing, posting, & sending data;
Method for solving hourly DC optimal power flow;
Methods for posting schedules and carrying out settlements;
Methods for implementing market power mitigation;

// Private Data

Historical data (e.g., cleared bids/offers, market prices,…);
Address book (communication links);

AMES ISO Solves Hourly DC Optimal Power Flow (OPF)

GenCos report hourly supply offers and LSEs report fixed & price-sensitive hourly demand bids to ISO for day-ahead market

Minimize

$$\sum_{i=1}^{I}[a_i^R p_{Gi} + b_i^R p_{Gi}^2] - \sum_{j=1}^{J}[c_j p_{Lj}^S - d_j {p_{Lj}^S}^2] + \pi \left[\sum_{km \in BR} [\delta_k - \delta_m]^2\right]$$

GenCo-reported total avoidable costs

LSE gross buyer surplus

$$\text{w.r.t.} \quad p_{Gi},\ i = 1, ..., I;\ p_{Lj}^S,\ j = 1, ..., J;\ \delta_k,\ k = 1, ..., K$$

Subject to

$$\sum_{i \in I_k} p_{Gi} - \sum_{j \in J_k} (p_{Lj}^F + p_{Lj}^S) - \sum_{km \text{ or } mk \in BR} B_{km} [\delta_k - \delta_m] = 0$$

$$|B_{km} [\delta_k - \delta_m]| \leq P_{km}^U$$

$$Cap_i^L \leq p_{Gi} \leq Cap_i^{RU}$$

$$0 \leq p_{Lj}^S \leq SLMax_j$$

$$\delta_1 = 0$$

Fixed and price-sensitive demand bids for LSE j

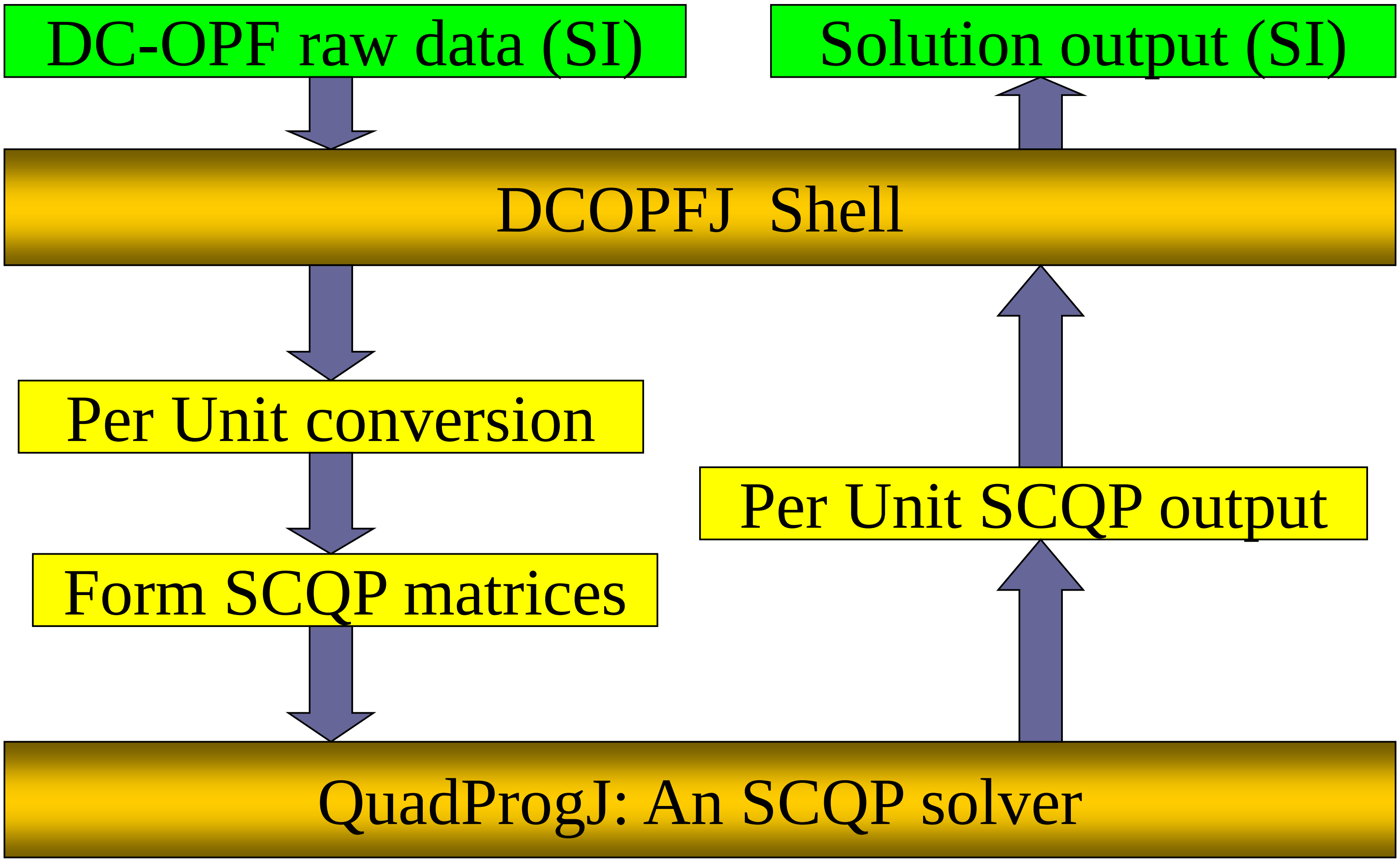
Dual variable for this bus-k balance constraint gives LMP for bus k

Operating capacity interval for GenCo i

Purchase capacity interval for LSE j

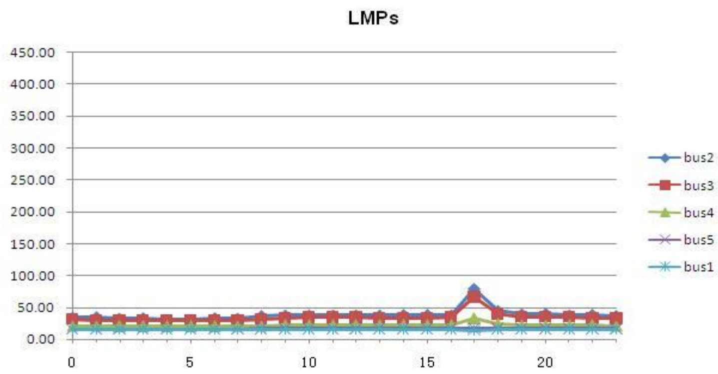
P^U_{km} = Upper limit on branch km

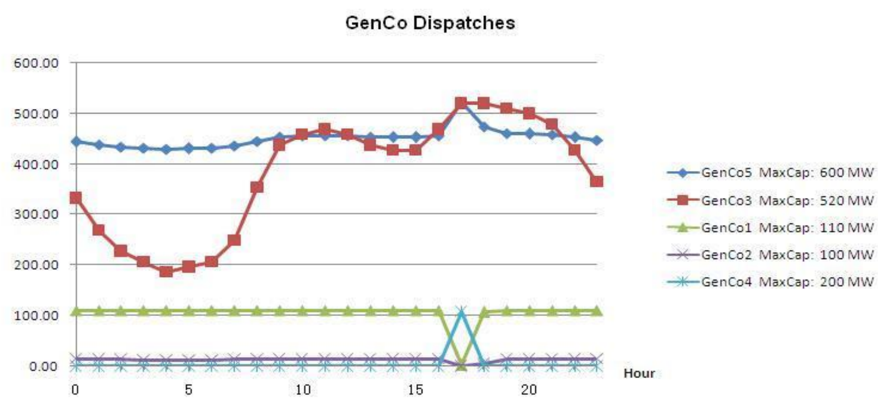
AMES ISO Solves DC-OPF via DCOPFJ Module



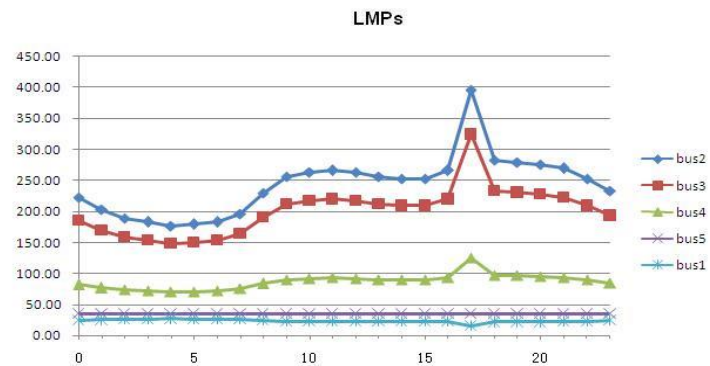
Illustrative 5-Bus Findings for R=0.0 (100% Fixed Demand)

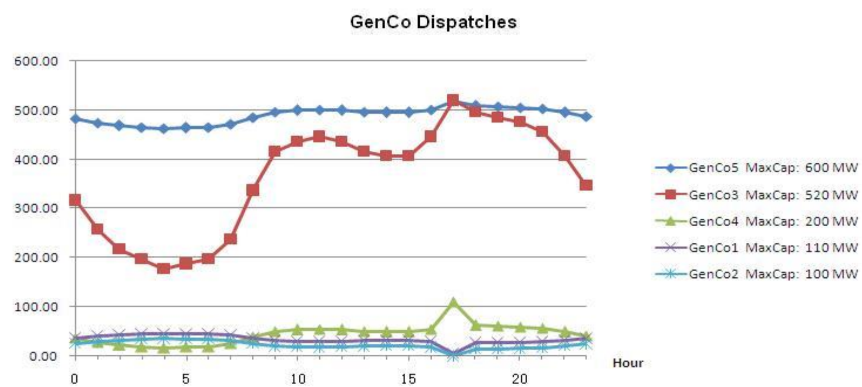
W/O Gen Learning (Day 1000)

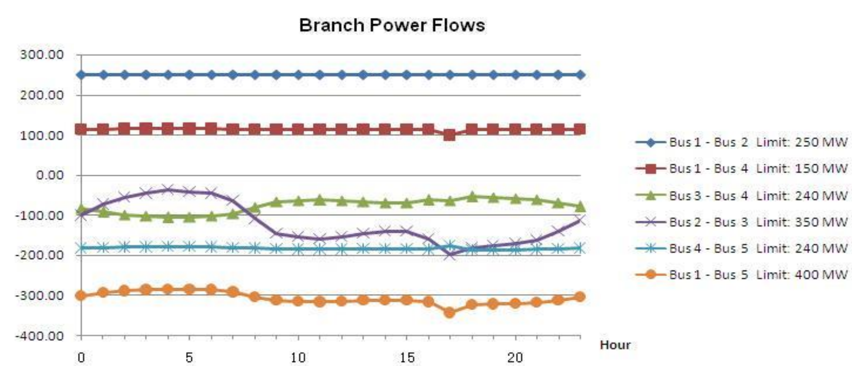




With Gen Learning (Day 1000)







Online Resources

- **Agent-Based Comp Econ Homepage**
  https://www2.econ.iastate.edu/tesfatsi/ace.htm
- **ACE/CAS Comp Labs and Demos**
  https://www2.econ.iastate.edu/tesfatsi/acedemos.htm
- **ACE/CAS: General Software & Toolkits**
  https://www2.econ.iastate.edu/tesfatsi/acecode.htm
- **AMES Wholesale Power Market Testbed**
  https://www2.econ.iastate.edu/tesfatsi/AMESMarketHome.htm
- **Agent-Based Electricity Research**
  https://www2.econ.iastate.edu/tesfati/aelect.htm