TUTORIAL: Programming agents in electronic institutions

IIIA-CSIC

Goal

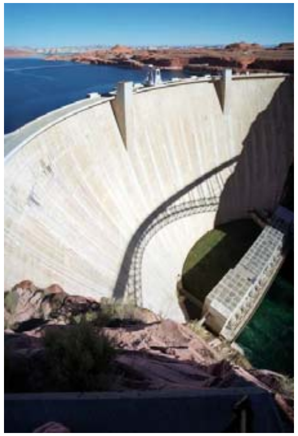
- To learn how to run electronic institutions over the AMELI middleware

- To learn how to simulate electronic institutions for what-if-analysis

- To provide a gentle introduction to agent programming in electronic institutions

- Case study: the double auction in the electricity market

Producers



Howt to build bidding strategies for external agents (producers and consumers) in the double auction (primary market)?

Consumers

Primary market

- Implemented as a **discriminatory double auction**.
- Each trader submits an offer price to the clearinghouse along with an offer quantity.
- The auctioneer separately sorts the buyers and sellers by their price offers in descending and ascending order.
- The buyer with the highest bid price is first matched with the seller with the lowest ask price.
- The unit price for the contract is set at the midpoint of the bid-ask spread.

Discriminatory double auction

TABLE I
BUYER-SELLER MATCHING ILLUSTRATION

Sellers	Buyers
$4 / 20 MWh;	$9 / 10 MWh
$5 / 10 MWh;	$8 / 10 MWh
$6 / 10 MWh;	$7 / 10 MWh

Matches: (1-1) for 10 MWh at Unit Price $6.50/MWh;
(1-2) for 10 MWh at Unit Price $6/MWh;
(2-3) for 10 MWh at Unit Price $6/MWh;
Seller 3 Not Matched.

Proposal I: SLIE producers

- To decide the amount of energy to be generated during the next half an hour, as well as the price to participate in the double auction of the next primary market, the producer uses the result of its participation in the previous primary market.
- If in the previous round of the primary market the producer was able to sell all his generated energy, he will increase his production a quantity equal to **PowerUp** and will increase the price of the energy, otherwise it will decrease his production by a quantity equal to **PowerDown** and will decrease the price of the energy.
- These agents offer all their available capacity.

Proposal I: SLIE producers

$$Q^t = \begin{cases} Q^{t-1} + PowerUp & lastSold = true \\ Q^{t-1} - PowerDown & lastSold = false \end{cases}$$

$$P^t = \begin{cases} P^{t-1} + \varepsilon^+ \cdot \frac{|P^{t-1} - AvP^{t-1}|}{AvP^{t-1}} & lastSold = true \\ P^{t-1} - \varepsilon^- \cdot \frac{|P^{t-1} - AvP^{t-1}|}{AvP^{t-1}} & lastSold = false \end{cases}$$

$$FinalP^t = \max(P^t, \text{Production\_cost})$$

Production_cost: cost to generate energy

$FinalP^t$: price at time t

AvP^{t-1}: price paid at time t-1

$\varepsilon+$ and $\varepsilon-$ are two constants particular to each producer

Proposal I: SLIE consumers

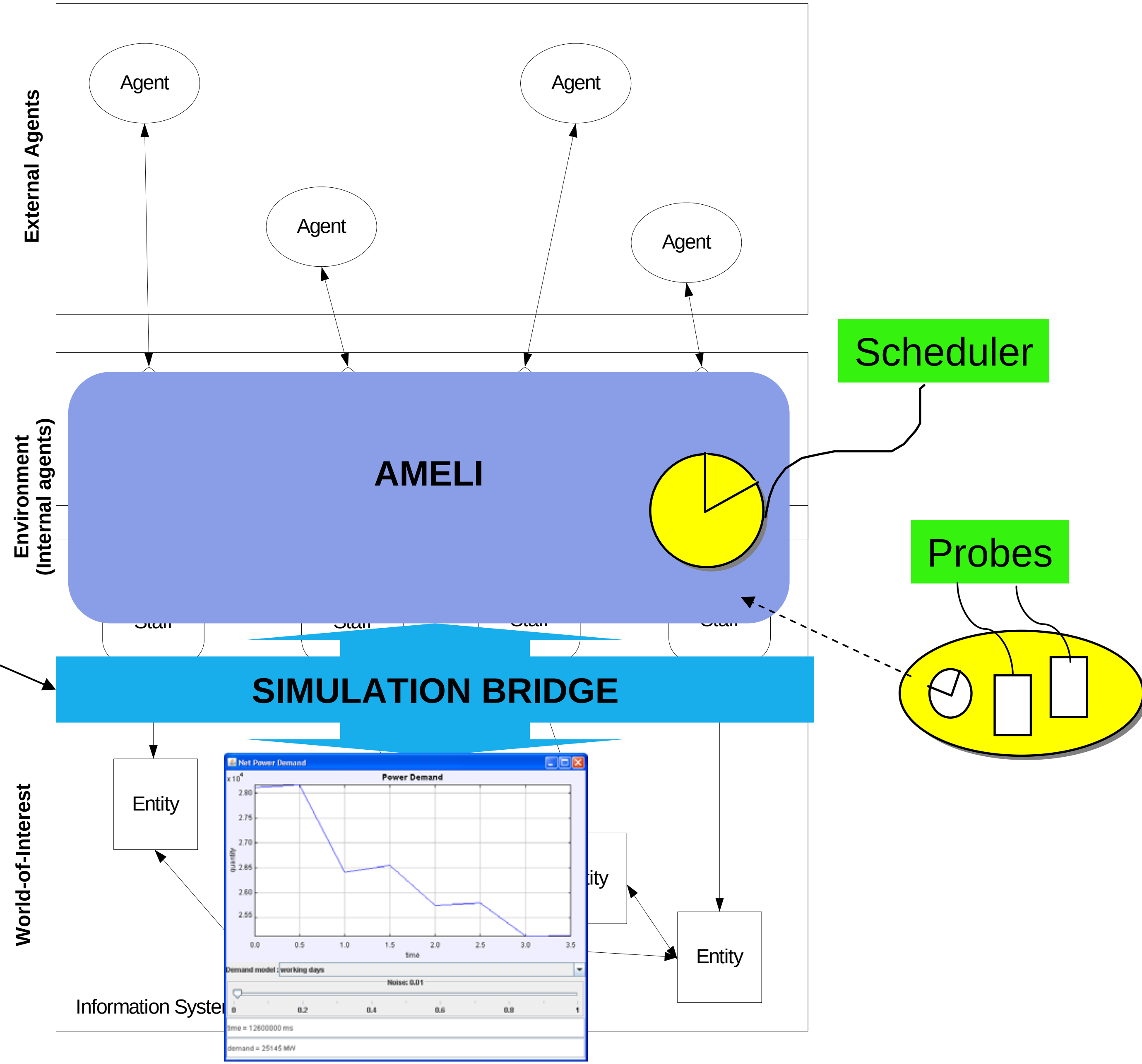
$$FinalP^t = \begin{cases} P^{t-1} - \varepsilon^+ \cdot \frac{|P^{t-1} - AvP^{t-1}|}{AvP^{t-1}} & lastBought = true \\ P^{t-1} + \varepsilon^- \cdot \frac{|P^{t-1} - AvP^{t-1}|}{AvP^{t-1}} & lastBought = false \end{cases}$$

The quantity of energy requested is equal to a market share (depending on each consumer) of the demand for the next half an hour. This quantity is obtained fromt a systems dynamic simulator (EJS) that provides consumers with a demand forecast service.

Demand forecast available to consumers

Functions

- Synchronise Simulations
- Event observation
- Event translation



Proposal II: Quantity-based producers

- Assess prices like SLIE agents
- Offer as much quantity as currently available as long as profit is guaranteed

$$q_i(P) = \begin{cases} CS_i, & \text{if } P > \alpha_i \\ [0, CS_i], & \text{if } P = \alpha_i \\ 0, & \text{if } P < \alpha_i \end{cases}$$

energy, αi stands for his cost

- Notice that this is a slight variation of SLIE producers.

Proposal III: Roth-Erev agents

- RE algorithm designed to play games with multiple strategically interacting players.
  - **Law of effect**: the tendency to implement an action should be reinforced if it produces favourable results and weakened otherwise
  - **Power law of practice**: learning curves tend to be initially steep, after which they flatten out
  - **Experimentation effect**: similar choices to successful choices in the past will be employed more often
  - **Recency effect**: recent experience plays a larger role than past experience.

Proposal III: Roth-Erev agents

- Nicolaisen et al. Adapt RE algorithm to the electricity market
- Given any feasible action k, the propensity for choosing k in the next auction round is

$$q_{jk}(n+1) = (1-r)q_{jk}(n) + E(j,k,k',n,K,e)$$

$$E(j,k,k',n,K,e) = \begin{cases} R(j,k',n)(1-e), & k = k' \\ q_{jk}(n)\frac{e}{K-1}, & k \neq k' \end{cases}$$

- And the choice probabilities for auction round n+1

$$p_{jk}(n+1) = \frac{q_{jk}(n+1)}{\sum_{m=1}^{K} q_{jm}(n+1)}.$$

- RE traders are **myopic**: given this profit outcome, what price should I next choose?

Proposal IV: Roth-Erev++ agents

- RE producers and consumers are not very aggressive. They stabilise around winning prices and slowly react to losses
- We propose to slightly modify RE agents as follows:
  - If an agent sells all his offered quantity, instead of reinforcing the winning price, he does reinforce a higher price because he considers the sale as an indicator of an increase in market demand.
  - If an agent does not manage to sell all his offered quantity, he reinforces lower prices because he considers that either: (i) the market demand is going down; or (ii) his competitors are going cheaper.
  - If an agent does not succeed in selling energy, he shifts the probabilities for the prices he has been using for winning prices to reinforce cheaper prices, and thus increase his probability of succeeding in the next round.

Proposal V: DIY agents

- Yes, indeed, do it yourself!!!