Software Agents and the Information Economy

Jeff Kephart
Agents and Emergent Phenomena
IBM Research

kephart@us.ibm.com

November 3, 2003
Two broad application areas

- **Information economy**
  - Large-scale competitive Multi-Agent System
  - Billions of economically motivated agents
    - Buying and selling information goods and services
    - Adaptive, and coupled directly and indirectly (through markets)

- **Autonomic computing**
  - Large-scale cooperative or competitive Multi-Agent System
  - Self-managing computing systems
    - Self-configuring, Self-healing, Self-optimizing, Self-protecting
    - The Vision of Autonomic Computing
      - IEEE Computer, January 2003
Outline

- **Information Economy**
  - Real software bidding agents vs. humans
  - Simulated bidding agents
  - Simulated pricing agents

- Autonomic computing

- Conclusions
Billions of interacting, adaptive agents. What emergent behaviors will arise?
Emergent Behavior

• Agent-agent interaction will increase over time
  – Direct interactions enabled by standards
  – Indirect interactions mediated by economy

• Agents are a new economic species
  – Fast idiot-savants
  – Their collective behavior may be very different
  – We need to understand likely modes of behavior

• Approach: model agent markets; study collective behavior
Outline

- Information Economy
  - Real software bidding agents vs. humans
    - Simulated bidding agents
    - Simulated pricing agents

- Autonomic computing

- Conclusions
CDA is common in financial markets

Extensive prior literature
- All-human experiments (Vernon Smith)
- All-agent experiments (SFI DA, Gode-Sunder, Cliff, TAC)
Limit Prices

Trade Summary

<table>
<thead>
<tr>
<th>Unit #</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ask</td>
<td>49</td>
<td>49</td>
<td>50</td>
<td>50</td>
<td>57</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Price</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cost</td>
<td>8</td>
<td>24</td>
<td>40</td>
<td>56</td>
<td>72</td>
<td>88</td>
<td>104</td>
<td></td>
</tr>
<tr>
<td>Profit</td>
<td>42</td>
<td>26</td>
<td>10</td>
<td>78</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Enter Asks

Ask per unit: [ ]
Number of units: [ ]

 Submit

Messages

* You have sold 1 unit at a price of 50 per unit
* You have sold 1 unit at a price of 50 per unit
* You have sold 1 unit at a price of 50 per unit

Market Transaction Prices

Ask Queue

<table>
<thead>
<tr>
<th>Ask</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>57</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ask</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>51</td>
<td>1</td>
</tr>
</tbody>
</table>

Bid Queue

<table>
<thead>
<tr>
<th>Bid</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>49</td>
<td>1</td>
</tr>
<tr>
<td>48</td>
<td>1</td>
</tr>
<tr>
<td>34</td>
<td>1</td>
</tr>
</tbody>
</table>
Bidding Agent Architecture

Auctioneer → Message Handler → Brain
- Wakeup?
- Compute Order
- Place Order

Auctioneer receives Auction Info and sends Orders.

Message Handler receives Auction Info and sends Orders.

Brain processes the messages and computes the order.

Bookkeeper processes the order and maintains Market state, Agent state, and Limit prices.

Agent GUI interacts with all components.
Agent-Human experiments

- **Human subjects**
  - recruited from local colleges and IBM Research
  - given interactive instructions and test
  - paid in proportion to surplus

- **Setup**
  - 6 Humans, 6 Agents
  - 6 Buyers, 6 Sellers
  - Each agent shares limit prices with a human

- **Experiment**
  - 9 to 16 3-minute periods
  - Limit prices change every 3-5 periods
  - Record bids, asks, trades
Experiment #6: Fast GD vs. Humans

6 GD Fast Agent-6 Human CDA Exp on Oct 25: Periods 1-16

Theor. Equilibrium Price
Agent-Agent
Agent-Human
Human-Human

Trade Price vs. Time (seconds)

PHASE 1
AB: 1.62
AS: 0.66
HB: 1.14
HS: 0.29

PHASE 2
AB: 1.51
AS: 0.62
HB: 1.15
HS: 0.49

PHASE 3
AB: 1.32
AS: 0.70
HB: 1.12
HS: 0.68

PHASE 4
AB: 0.99
AS: 1.01
HB: 0.89
HS: 1.01
Summary of experimental results

- Agents won by substantial margins in all experiments
  - ~20% more surplus than novice humans
  - ~5-7% more surplus than experienced humans

- Agents and humans interact with one another
  - Not two decoupled markets
  - ~30-50% of trades are agent-human

- Market efficiency improves with number of agents
  - Humans fare better when there are more agents

- Agents can supplant humans as economic decision makers
Outline

- Information Economy
  - Real software bidding agents vs. humans
  - **Simulated bidding agents**
  - Simulated pricing agents

- Autonomic computing

- Conclusions
All-agent experiments

- **Simulator**
  - Discrete-time; stochastic asynchronous dynamics.
  - Ran mixtures of several strategies and variants

- **Market parameters**
  - 10 buyers, 10 sellers
  - 10 units each. Fixed limit prices (chosen randomly)
  - 100 expts, 5 trading pds/expt, 300 time steps/pd.

- **Experimental comparisons**
  - Homogeneous (0 A vs. 20 B)
  - One-in-Many Tests (1 A vs. 19 B)
  - Balanced Team Tests (10 A vs. 10 B)
Expt. 2: 1 A vs. 19 B
Differential Efficiency

<table>
<thead>
<tr>
<th>Scale</th>
<th>ZI</th>
<th>Kaplan</th>
<th>ZIP</th>
<th>GD</th>
<th>MGD</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ 5%</td>
<td>-</td>
<td>□</td>
<td>□</td>
<td>-</td>
<td>□</td>
</tr>
<tr>
<td>- 5%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

- ZIP and MGD invade ZI, Kaplan & GD
  - But don’t invade one another
- Kaplan can invade all strategies
- All strategies invade ZI
- ZI doesn’t invade any
Expt. 3: 10 A vs. 10 B
Differential Surplus (out of 2612 total)

<table>
<thead>
<tr>
<th>TEAM A</th>
<th>Scale</th>
<th>Team B</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>= + 500 units</td>
<td>ZI Kaplan ZIP</td>
</tr>
<tr>
<td>Kaplan</td>
<td>= - 500 units</td>
<td></td>
</tr>
<tr>
<td>ZIP</td>
<td>(0-100)</td>
<td>(100-0) (99-1)</td>
</tr>
<tr>
<td>GD</td>
<td>(99-1)</td>
<td>(93-7) (36-63)</td>
</tr>
<tr>
<td>MGD</td>
<td>(99-1)</td>
<td>(98-2) (71-29)</td>
</tr>
</tbody>
</table>

- **ZI beats** Kaplan 100-0!
- Other strategies beat Kaplan, but by smaller margin
- **GDX > MGD > ZIP > GD > ZI > Kaplan**
Kaplan vs. ZI

1 Kaplan invades 19 ZI

19 Kaplans resist 1 ZI

In between, $20-n$ ZI beat $n$ Kaplan
Evolutionary dynamics
CDA game

What happens when agents gradually switch to more successful strategies?

No strategy is dominant.

This is a useful view for
•Mechanism design
•Agent design
Outline

- Information Economy
  - Real software bidding agents vs. humans
  - Simulated bidding agents
    - Simulated pricing agents

- Autonomic computing

- Conclusions
Shopbots and pricebots

With Amy Greenwald
Brown University

Amazon.com  Borders.com  bn.com

GT  DF  MY

pricebots
(various pricing algorithms)

shopbot

users/user agents
(various buyer strategies)
Profit landscape and best-response policy
Shopbots and pricebots
Dynamic pricing game
Shopbots and pricebots

Time series: cyclical price war
Cyclical price wars
Price/Quality Model

5 myoptimal sellers

5 sellers; vertically-differentiated good
Cyclical price wars
Information Bundling Model

5 sellers; multi-attribute good

5 sellers, 3 categories
Valuations: Uniform[0,1]
Cyclical price wars
Underlying cause

• **Existence of profit landscape**
  – Consistent relationship between price and profit
  – Responsive buyer agents support this by reducing inertia

• **Multiple peaks in profit landscape**
  – Very easy: sharp cliffs occur when $p_1 = p_2$

• **Global price exploration**
  – Global optimization of known landscape
  – Randomized exploration with occasional large jumps
Dynamic Pricing
Mix of strategies

Nash
Seller optimal
Meta-Payoff Matrix
Evolutionary dynamics
Dynamic pricing game

Nash Computation

<table>
<thead>
<tr>
<th>Agents</th>
<th>Label</th>
<th>p(GT)</th>
<th>p(DF)</th>
<th>p(NIR)</th>
<th>Payoff</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>A</td>
<td>1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.051</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>0.871</td>
<td>0.000</td>
<td>0.129</td>
<td>0.049</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>0.030</td>
<td>0.000</td>
<td>0.969</td>
<td>0.047</td>
</tr>
<tr>
<td>20</td>
<td>D</td>
<td>0.986</td>
<td>0.000</td>
<td>0.014</td>
<td>0.013</td>
</tr>
</tbody>
</table>

5 Agents (a)

20 Agents (b)
Information Economy

Summary

- Several models
  - CDA bidding model
  - Shopbot/pricebot dynamic pricing models
    - Single undifferentiated good
    - Vertically differentiated good (price/quality)
    - Horizontally differentiated goods (information bundling)

- ABMS approach
  - Each agent characterized by plausible strategy/algorithm
  - Simulate/analyze collective market behavior
  - Gain insights into
    - Root cause of interesting dynamic behavior (e.g. price war cycles)
    - Design of effective individual algorithms
    - Collective strategic interactions
      - Design of individual agents; market mechanisms
Outline

- **Information Economy**
  - Real software bidding agents vs. humans
  - Simulated bidding agents
  - Simulated pricing agents

- **Autonomic computing**

- Conclusions
Complex heterogeneous infrastructures are a reality!

Dozens of systems and applications

Hundreds of components

Thousands of tuning parameters
Autonomic Computing
Self-managing computing systems

- Administration of individual systems is increasingly difficult
  - 100s of configuration, tuning parameters for DB2, WebSphere

- Heterogeneous systems are becoming increasingly connected
  - Integration becoming ever more difficult

- Architects can't intricately plan interactions among components
  - Increasingly dynamic; more frequently with unanticipated components

- More of the burden must be assumed at run time
  - But human system administrators can't assume the burden; already
    - 6:1 cost ratio between storage admin and storage
    - 40% outages due to operator error

- We need self-managing computing systems
  - Behavior specified by sys admins via high-level policies
  - System and its components figure out how to carry out policies
Evolving towards Self-management

<table>
<thead>
<tr>
<th></th>
<th>Today</th>
<th>The Autonomic Future</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Self-configure</strong></td>
<td>Corporate data centers are multi-vendor, multi-platform. Installing, configuring, integrating systems is time-consuming, error-prone.</td>
<td>Automated configuration of components, systems according to high-level policies; rest of system adjusts automatically. Seamless, like adding new cell to body or new individual to population.</td>
</tr>
<tr>
<td><strong>Self-heal</strong></td>
<td>Problem determination in large, complex systems can take a team of programmers weeks</td>
<td>Automated detection, diagnosis, and repair of localized software/hardware problems.</td>
</tr>
<tr>
<td><strong>Self-optimize</strong></td>
<td>WebSphere, DB2 have hundreds of nonlinear tuning parameters; many new ones with each release.</td>
<td>Components and systems will continually seek opportunities to improve their own performance and efficiency.</td>
</tr>
<tr>
<td><strong>Self-protect</strong></td>
<td>Manual detection and recovery from attacks and cascading failures.</td>
<td>Automated defense against malicious attacks or cascading failures; use early warning to anticipate and prevent system-wide failures.</td>
</tr>
</tbody>
</table>
Autonomic Computing Architecture
The Autonomic Element

- AEs are the basic atoms of autonomic systems

- An AE contains
  - Exactly one **autonomic manager**
  - Zero or more **managed element(s)**

- AE is responsible for
  - Managing own behavior in accordance with policies
  - Interacting with other autonomic elements to provide or consume computational services

**Service-oriented architecture**

**Software agents**

E.g. Database, storage, server, software app, workload mgr, sentinel, arbiter, OGSA infrastructure elements
Autonomic Computing Architecture
Element interactions

- System self-* properties, behavior arise from interactions among autonomic managers

- Interactions are
  - Dynamic, ephemeral
  - Formed by (negotiated) agreement
  - Flexible in pattern; determined by policies
  - Based on OGSA and specific AC extensions
    - Required messages
    - Optional but standard
    - Application-specific

- For advanced interactions: conversation support
  - “Choreography” defines structure of multi-step interactions

A multi-agent system!
Example scenario: Autonomic Data Center
Outline

- Information Economy
  - Real software bidding agents vs. humans
  - Simulated bidding agents
  - Simulated pricing agents

- Autonomic computing

- Conclusions
Closing remarks

- Two application areas
  - E-commerce: competitive, giga-agent MAS
  - Autonomic computing: cooperative or competitive MAS
      - IEEE Computer, January 2003
      - http://www.autonomic-conference.org

- We are building real prototype agents
- We are using ABMS to simulate these “real” agents
  - Observations
    - Differences from observed human behavior (e.g. price wars)
    - Interesting strategic interactions; analyze using meta-payoffs
  - Insights can help design
    - individual agents (new for economics!)
    - interaction mechanisms
    - systems