An Agent Based Value Chain Coordination Experiment

Torsten Eymann, Boris Padovan, Stefan Sackmann
Institut für Informatik und Gesellschaft (IIG), Abteilung Telematik
Albert-Ludwigs-Universität Freiburg, Germany
{eymann|padovan|sackmann}@iig.uni-freiburg.de
Introducing Autonomous Agents and Multi-Agent Systems

An **agent** is software that **acts pro-actively** and **autonomously** on behalf of a (human) principal. Agents work by allowing people to **delegate** work that they could have done, to the agent software. A **multi agent system (MAS)** provides a global problem solution by local interaction of the distributed agents.
The complex, networked reality

Examples:
Electronic Markets, Production Planning, Traffic Control, Power Grid Control, Climate Control in Buildings, Distributed Systems Design, Transportation Scheduling, Supply Chain Management

Characteristics:
- many, many individualistic participants
- lateral co-operation, but no benevolence
- No organizational authority to enforce rules
- computationally NP-hard problems of resource allocation
Application Scenario: A B2B electronic marketplace

I want to buy one table for $120, maybe $150 if I really like it

I want to buy wood for $35 or less and sell boards for $60 or more

I want to buy boards for $50 or less and sell tables for $120 or more

I want to sell wood for $35 or more
The Research Challenge

- Co-ordinate a (virtual enterprise) supply chain via a B2B market
  - consisting of self-interested participants of unknown reputation
  - without participants having to reveal strategies or utility functions to anybody
  - with a real time solution (no turns) for resource allocation
  - with decisions based on structurally incomplete information (constitutional ignorance)
  - without a bottleneck or single point of failure (scalability!)
  - with fast and flexible creation and dissolution of co-operational links (no “stable systems drag”)
The Research Process

- **Do Implementation**
  - Build a realistic electronic marketplace
    - scalable, robust, networked, flexible, easy to understand
    - with decentralized co-ordination
  - Create supply chain participants with individualistic cooperation strategies
    - Humans or software agents
    - autonomous decision on co-operational behaviour
  - Show what happens in the longer run
    - Will the supply chain deliver?
    - Which strategies prevail?
    - What global behaviour occurs?

- **Do Simulation** (Computational Organization Theory):
  - What is the impact of changes in
    - trade strategies of single agents
      (transaction costs, co-operational behaviour)
    - the economic environment
      (transparency of supply and demand, connectivity of marketplaces)
  - on the stability of the supply chains raised?
    - short, brittle, unstable chains = market-like coordination
    - long, durable, stable chains = organisation-like coordination
The Avalanche Environment - A Supply Chain Management Setup

Wood → Board → Plate → Table

Lumberjack Agents → Carpenter Agents → CabinetMaker Agents
Key elements of the implementation

- **Agent technology**: an open environment where people can bring in their own software entities
  - Multi-agent technology standards where applicable, no proprietary solutions: FIPA, CORBA, X.509, Java
- **Economic reasoning**: A (feedback) mechanism to guide people’s decisions
  - Derived from the “invisible hand” of economics
  - But how do I implement a “market”?
- **Evolution**: An adaptation process to improve the solution to resource allocation over time
  - Derived from Evolutionary Algorithms
  - But what is an acknowledged fitness indicator?
  - How can evolution be justified in economic terms?
Economic Agents = Software Agents, implemented on top of Objectspace’s Voyager Distributed Systems Middleware

Marketplaces (Locations) = Java Virtual Machines on IP Ports with Directory functionality (passive Yellow Pages, only agent ID’s)

Automated Negotiation between Agents based on FIPA’s Agent Communications Language (iterated-contract-net protocol)
Implementation: System Architecture

AvTradeAgentProperties.class
Genotype.class
AvTradeAgent.class
AvLocationAgent.class
AvInfoServer.class

Trading Mating
Advertise

Logging Display Control

Yellow Pages Service
Incubator Queue
Timestamping
Central Bank Service
Show Avalanche Demo Now
Long-term market behaviour

- Every tick marks a completed deal
- Agents can be considered as trading in-between two curves
Economic Reasoning: Agent Design
Properties of an Avalanche agent

production properties
technology (x materials → y goods)
materials price
goods price
production cost

negotiation properties (genotype)
Acquisitiveness
In-Negotiation Price Change
Start-Price Change
Negotiation Progress Satisfaction
Market Price Memory Weight
Agent Reputation

pro-active behaviour driver
local goal: maximize income
(sell higher than last deal, buy lower than last deal)

cashflow properties
capital / cash
costs of living% (taxes)
How the Avalanche agent works

- **Products available?**
  - Yes → Advertise products or change offer price
  - No → Materials available?
    - Yes → Production: Change materials to goods
    - No → Search sellers, do ranking
      - Satisfied with market situation?
        - Yes → Buy materials
        - No → Change Location
          - Subtract costs of living from capital

© 2000, IIG Telematik, Albert-Ludwigs-Universität Freiburg, Germany
Negotiation Implementation (1)

- **Communication** using FIPA's Agent Communication Language
- **Cooperation** using a ACL "iterated contract net"-like protocol
  - Bilateral haggling with opt-out option
  - Decreasing seller offers
  - Increasing buyer bids
Negotiation Implementation (2)

- Bilateral haggling with probabilistic concessions
- Opponent’s decision is non-deterministic

- Negotiation parameters
  - p_satisfaction: Should I leave this negotiation?
  - p_acquisitiveness: Should I change my bid price?
  - del_change: How big should the change in my bid price be?
  - + 3 more
Negotiation Implementation (3)

Example Parameters:

- Acquisitiveness \( p_{acq} \) = 0.7
  - 70% probability to stick with the own offer

- Delta Price Change \( \text{del}_\text{change} \) = 0.5
  - make a concession by offering last price +/- \( \text{del}_\text{change} \times \text{spread} \)

- \( \Delta s = (p_s - p_B) \times \text{del}_\text{change} = (24-18) \times 0.5 = 3 \)

- \( p_s = \begin{cases} RND < p_{acq} : p_s^t = p_s^{t-1} \\ RND \geq p_{acq} : p_s^t = p_s^{t-1} - \Delta s \end{cases} \)
Negotiation Implementation (4): The buyer agent as a non-deterministic FSM

input/probability/output
p = offer.price
mp = floating average of estimated marketprice
Negotiation Implementation (4): intelligent agents?

- Source: k-level concept of Vidal/Durfee (AAAI 1996)
- 0-level agents have no internal model of the environment
  - reactive agents, ZIP agents, stigmergic agents, alife
  - current Avalanche
- 1-level agents resolve their environment to be 0-level agents
  - Avalanche with reputation vector
  - selective choice of transaction partners
- 2-level agents resolve their environment to be 1-level agents
  - from here on: AI stuff
- and so forth
Evolutionary development of agents = strategies

Fitness selector: floating average on profits realised from last sales (sellPrice - stockItemPrice)
Why Evolution?

- No economic rationale for genetic algorithms
  - In economics, there is no mating, crossover or mutation
  - A standing problem for „Evolutionary Economics“

Working rationale

- Initial result on resource allocation depends on starting set of agents (= strategies)
- „Evolutionary algorithm“ attempts to optimize the solution
  - Bad strategies/agents are removed from the marketplace (going bankrupt)
  - Good strategies/agents increase (new market contestants imitate successful ones)
- EA is an inherent part of the market and just a question of implementation
Simulation Results
Acquisitiveness 80%  
Development of goods prices

- Few trades because one seldom gives in to a compromise
- Homogeneous strategies
- No development because all agents are equal and no variation allowed in the $p_{acq}$ parameter
Acquisitiveness 60% with a possible Variation of 0.9

- More trades because of faster compromising
- Blurred price levels because of the variation of $p_{acq}$
- Which acquisitiveness setting will prevail?
Acquisitiveness 60% with a possible Variation of 0.9

- Graph shows average $p_{acq}$ for each agent type whenever a new agent is created
- Evolutionary development of the $p_{acq}$ parameter
- Convergence dependent on other agent’s settings
Acquisitiveness of Carpenters
60%, others 40%

Graph shows average \( p_{acq} \) for each agent type (Initial variation was 0.4)

- Carpenters are able to relax, others have to get more acquisitive
- How will the prices develop?
Carpenters are able to cut the prices in their favour.
Later on, the lumberjacks gain some room by sacrificing the producer agents.
Acquisitiveness of Carpenters
60%, others 50%

- Faster convergence since the initial $p_{acq}$ spread is not so big
- How will the prices develop now?
Acquisitiveness of Carpenters
60%, others 50%

- Price levels change subject to dynamic adaptation in the agents
- In recent simulations, the system may run like that for hours
- In other runs, it may fail immediately
About the Design of Coordination Mechanisms

What is that kind of coordination good for?
Isn’t it easier to calculate the allocation beforehand?

Is there an economic theoretical body behind our concept?
A Layered Framework for Multi-Agent Systems

Application

Coordination

Cooperation

Communication

Application Services
Network Services
Physical Services

Wood

Board

Plate

Table

Lumberjack Agents

Carpenter Agents

CabinetMaker Agents
How co-ordination and markets relate

- A coordination mechanism is a method for resource allocation
- Resource allocation in markets is a NP-complete problem

- Solutions for NP-complete problems are either
  - heuristic (see production scheduling) or
  - parallel (see market-based agent concepts and auctions)

- Avalanche is a parallel solution to resource allocation
  - A formalized market implementation ...
  - … with concurrent computation and application of the solution
  - … with adaptation to a priori unknown changes
    - 1) Action decision depending on „market price“ and subjective valuation
    - 2) „Evolutionary algorithm“ continually attempts to maximize profits
## What is wrong with a central problem solver (arbiter?)

<table>
<thead>
<tr>
<th>Goal</th>
<th>Market with central arbiter</th>
<th>Decentralised Market</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fast calculation</td>
<td>Heuristic solution depending on designer’s knowledge</td>
<td>Parallel solution</td>
</tr>
<tr>
<td>Cheap calculation</td>
<td>Arbiter has to be paid by the participants</td>
<td>No overhead costs</td>
</tr>
<tr>
<td>Feedback to the participants on the success of their actions</td>
<td>Arbiter has to send explicit information about the solution quality</td>
<td>Information is gained by comparing own performance over time</td>
</tr>
<tr>
<td>Autonomous decision-making and communication</td>
<td>Utility and/or preference functions have to be revealed to the arbiter</td>
<td>Communication consists of bid prices</td>
</tr>
<tr>
<td>ad-hoc establishment and dissolution of cooperation links</td>
<td>Participants have to wait for the arbiter’s actions (bottleneck)</td>
<td>No concept of stable cooperation links</td>
</tr>
<tr>
<td>Resolution of conflicts</td>
<td>Arbiter has to discover and resolve conflicts computationally</td>
<td>Conflicts are mirrored in prices and resolved by bidding strategies</td>
</tr>
<tr>
<td>Stability of solution</td>
<td>Solution holds until the next round of computation</td>
<td>Solution is void the moment any deal has been made</td>
</tr>
</tbody>
</table>
Co-ordination: constructed or emergent?

Distributed Problem Solving

- Global solution
  - Intm. solution
    - Local solution
    - Local solution
    - Local solution

Multi-Agent Systems

- Global solution
  - Intm. solution
    - Local solution
    - Local solution
    - Local solution

Identification of a single best solution
Hierarchical, top-down deconstruction
Local solution by autonomous problem solvers (Agents)
Defined composition of results

Local solution by autonomous problem solvers (Agents)
Bottom-up, emergent composition of results
How to „grow“ emergence?
How to realise technically?
An economic view on co-ordination

- “Fundamentally, in a system in which the knowledge of the relevant facts is dispersed among many people, prices can act to co-ordinate the separate actions of different people in the same way as subjective values help the individual to co-ordinate the parts of his plan.” (Friedrich A. von Hayek, The Use of Knowledge in Society, 1945)

- Hayek’s Notion of the “Catallaxy”
  - agents work in their own interest to gain income
  - agents subjectively weigh and choose preferred alternatives
  - agents have access to markets which provide feedback
  - agents are subject to “constitutional ignorance”
  - system needs a legal framework (institutions)

- Avalanche as a prototype for “Catallactic Information Systems”
- A “libertarian” paradigm for IS design
Visions and Applications

What can this concept be applied to?
What is it good?
Application Domains for Avalanche’s Concepts

- Networked systems made of self-interested participants where
  - Trade items are commodities
    - or based on a specialized ontology (e.g. EAN-Code)
    - so we do not need a phase on negotiating on what to negotiate about
  - Interactions are numerous
    - the more deals are made, the better the price mechanism contributes
    - means: fast, often, repeated deals on small chunks of items
- Proposed application areas
  - e.g. communication bandwidth, power exchanges, low-value procurement, resource allocation with ubiquitous computing devices
Avalanche as a Prototype for Market-Based Process Management

- Key Idea: Subsequent markets are the most efficient way of coordinating any kind of process

- Caveat: Markets are Complex Dynamic Systems with no a-priori-guarantee what the outcome will be

- Software Agent Traders affect each other's behaviour via the impact of their actions on prices and availability

- Complex emergent behaviour arises from the cumulative effects

- possible negative outcome: boom-and-bust cycles (Parunak) price wars (Kephart)

- possible positive outcome: self-organisation, self-regulation, adaptation
Scope and Research Challenges of Market-Based Process Management

1. Diagnose and cure aberrant behaviour...
   - in agents by protecting property
     - strong security,
     - continuous adaptation,
     - risk reconnaissance
   - in markets by enforcing contracts
     - create regulatory institutions
     - promote reputation tracking
     - provide laws, tariffs, taxes
   AmEC SIG
   ABSS SIG

2. Develop theoretical and empirical principles and techniques for specification of market-based systems with desirable behaviour
   MACC SIG
   SE SIG

3. Develop software modelling tools and components for agent-based “emergent computation”
   SE SIG

© 2000, IIG Telematik, Albert-Ludwigs-Universität Freiburg, Germany