Multi-Agent Technology: An Introduction

Professor George Rzevski
Professor Emeritus, Design and Innovation Department, The Open University, Milton Keynes
Founder & Chief Scientist Magenta Corporation Ltd, London

The Storyline

1. With the emergence of the global, Internet-based economy the complexity of our business, administrative, health, education and social activities is increasing at a considerable rate.

2. Conventional computer systems supporting these activities are too rigid to cope with high levels of complexity.

3. Complex activities can be effectively supported only with adaptive systems

4. Complex problems require complex problem solving tools.

5. The key property of adaptive systems and complex problem solving tools is Intelligence.

6. Distributed intelligence is a particularly suitable tool for supporting complex problem solving.

7. Multi-agent technology is currently the most cost-effective technology for creating complex problem solving tools.

Educational Objectives

These lecture notes should help the reader, in general, to understand fundamentals of modern science of complexity; to appreciate its importance; to begin to think in terms of complexity concepts and principles; to grasp a methodology for solving complex problems and to appreciate the power of multi-agent technology; and in particular, to

- Describe main features of complexity
- List examples of complex systems from physics, chemistry, biology, business, administration, engineering and computing
- Explain sources of complexity in business and administrative systems
- Describe how intelligence can help solving complex problems
- Explain why resource allocation problems are complex
- List at least ten complex resource allocation problems
- Appreciate roles of what-if? analysis, design, planning and real-time scheduling of complex systems
- Describe a method for solving complex problems
- Build virtual worlds
- Explain the concept of multi-agent technology to an intelligent layman
- Select complex problems that can be solved using multi-agent technology
- Select a multi-agent tool suitable for solving a particular complex problem

Introduction

These lecture notes do not provide an overview of the subject area of multi-agent technology (MAT). They reflect a particular way of thinking about agents and present a unique approach to the development of multi-agent systems. What follows are largely my own definitions and explanations which I derived from my research, design and application experience.

I have developed this particular approach to agent-based technology back in the early 1980s starting with the assumption that as we enter into global, Internet-based economy our business, administrative, educational, health-care and social activities will be conducted in a highly volatile, dynamic environments and therefore these activities will need to be supported by systems that can rapidly react to the occurrence of unpredictable events and co-evolve with their changing environments.

Conventional information systems are not effective in volatile business and social environments because they are purely algorithmic, based on logic, and therefore not capable of self-organisation or evolution.

Central to this approach to MAT has been the modern notion of complexity, as summarised below.

Complexity

During the last decades of the 20th century a considerable success was achieved in the development of Science of Complexity. Some of the basic concepts and principles of this new science are reviewed below. Readers are advised to make an effort and read books on the subject suggested in section Further Reading.

Let us start with a simple definition of Complexity.

A problem is said to be complex if it consists of a large number of varied elements and if it frequently and unpredictably changes.

And with a definition of Complex Systems:

A system is said to be complex if it consists of autonomous units (players, actors, agents) engaged in mutual interaction.

Scale, Variety and Uncertainty are key features of complexity.

For example, a transportation problem requiring 2000 orders a day to be allocated to 50 trucks (some large, some small, some with trailers and some without) distributed over a large geographical area crisscrossed by roads, where orders change hourly, and were roads may or may not be jammed, is a complex problem. In contrast, if a resource allocation problem consists of only two demands and three resources that need to be matched, it is not considered complex even if it frequently changes.

Complexity is in evidence in physical, chemical, biological and human-activity systems in which there are many autonomous elements engaged in rich mutual interaction.

Important characteristics of complex systems are:

- *Emergence* – the system exhibits properties that emerge from the interaction of its constituent elements and which are not in evidence in any of its constituent components. The secret of emergent behaviour is the existence of a set of norms which constituent agents are obliged to follow. If these norms are so comprehensive that agents have no freedom, the overall system behaviour is deterministic; if there are no norms, the system behaviour is random. Somewhere in between, when norms are such that agents have freedom to be pro-active and creative, the system will exhibit emergent behaviour. Typically, complex systems have no centralised control and yet they tend to produce coherent emergent behaviour.

- *Selforganisation* – the system is capable of autonomously (without being instructed) changing its own organisation in response to internal or external events. Systems capable of self-organisation are said to be Adaptive.

- *Evolution* – the system is capable of autonomously adjusting to changes in its environment over time. In fact a complex system *Co-evolves* with its environment – environment changes the system and changed system changes its environment.

Complex problems are unpredictable processes – they are never static or repeatable. Therefore there cannot be the optimal solution to a complex problem. The optimum changes as the problem evolves, which could be as frequently as every few hours.

Examples of complex systems include:

- Molecules of air subjected to a heat input; autocatalytic chemical processes; self-reproduction of cells; brain
- Colonies of ants; swarms of bees; ecology
- Cities; human communities; epidemics; terrorist networks
- Free market; global economy; supply chains; business processes; production processes; teams
- Multi-agent systems and possibly the Internet

To understand complexity it is helpful to place complex systems in the context of a general system classification.

RANDOM SYSTEMS (total chaos)	**COMPLEX SYSTEMS** (systems far from equilibrium)	**EQUILIBRIUM SYSTEMS** (steady-state, stable)	**ALGORITHMS** clocks
Max uncertainty No norms of behaviour	Considerable uncertainty; Flexible norms of behaviour	Negligible uncertainty; Elaborate rules of behaviour	No uncertainty; Every step prescribed
Random behaviour	Emergent behaviour	Planned behaviour	Deterministic behaviour
Disorganised	Self-organising	Organised and controlled	Rigidly structured with no possibility of deviation
Random changes	Co-evolving with their environments	No changes allowed; Small deviations from equilibrium possible	No changes of any kind possible

A system classification

Complexity is of particular importance to us today as we find ourselves in transition from the stable Industrial Economy to volatile, dynamic Information Economy.

Complexity and the Internet-Based Global Economy

We are in transition from a set of national industrial economies to the highly interconnected Internet-based Global Economy with radically different market conditions. As more countries join the competition, a considerable increase in manufacturing and, to a lesser extend, in service capacities is created. Demand patterns change frequently and in an unpredictable manner. Customers tend to prefer highly customised products and services and switch frequently from one supplier to another, searching for the best deals. In fact, the occurrence of unpredictable changes in orders is so frequent that automated production systems cope with difficulty, if at all. The economy of scale is now often counter-productive. When the frequency of changes in orders is high and the production line is rigid (expensive to change), the large production runs generate large numbers of unwanted products.

Automation (repetitions of predictable actions) is a good example of rigid, logic-based systems which were very successful in stable conditions of Industrial Economy. The automation is a cost-effective way of executing a complicated activity (such as a car production) but only if it is possible to decompose it into simple elementary activities, each executable by a precisely specified rule or algorithm, and if it is practical to eliminate or, at least, significantly reduce the occurrence of unpredictable events. In the past the accepted approach was to neglect complexity of problems by ignoring their dynamics and uncertainty for a certain limited time span and then applying deterministic methods. For example, during the industrial era it was acceptable to assume that the demand for certain car model will be stable for a period of 5-7 years and then build a rigid production line which manufactures a planned number of vehicles for a planned number of years. Similarly, pipeline-based logistic was very effective whenever the time for a pass through a pipeline was less than the time between changes in orders. Business processes were (and in many cases, still are) rigidly prescribed and employee were given lengthy instructions how to handle every foreseeable business transaction. With the emergence of global economy dynamics of markets increased to such extend that these approaches became counter productive.

The Economy of Scale worked wonders for automation – the larger the production plant, the cheaper the product. To increase cost-effectiveness of automated systems the idea was to make them Lean, that is, without any non-productive elements. Lean and automated production systems, which generated so much wealth under stable demand and supply conditions that we enjoyed until recently, is proving to be inadequate in turbulent market situations that are currently a norm. What is important under current market conditions is Adaptation, ie, responsiveness to changes in demand and supply, rather than the size of the system. We need the agility, that is, the ability to rapidly (1) decide how to adjust production or service facilities in response to market fluctuations and (2) implement selected changes. Thus, the new principle of economic production is the Economy of Governance, rather than the Economy of Scale. When the frequency of changes in orders is high, the greater the adaptability (the ability to adjust to changes), the less costly is the product.

Complexity and Intelligence

Conventional computer systems react to input data in a prescribed manner and when faced with uncertainty, that is, when they receive input data outside the allowed range, they stall, because they do not have instructions what to do with out-of range data or, if they are well protected, they may simply ignore unforeseen inputs.

If it is inadequate to support dynamic, unpredictable processes with rigid, logic-based systems, or to use such systems as tools for solving complex problems, then what kind of system or tool is required?

The answer is that complex processes can be supported only by complex systems of the same or similar complexity; and complex problems can be solved only with tools of the same or similar complexity.

Complex systems exhibit behaviour that is remarkably similar to intelligent behaviour of a swarm of bees or even of a group of humans - they adapt to changes and they are resilient to attacks without waiting to be instructed what to do. Very complex systems can achieve their goals under conditions of uncertainty and therefore they can be described as intelligent.

For the purposes of this notes I shall define *Intelligence*, in operational terms (by what it can do), as *the ability to achieve goals under conditions of uncertainty.*

In general intelligence is of course a very complex property. We shall cover below certain aspects of intelligence more broadly.

Aspects of intelligence which enable it to handle uncertainty include:

- Making choices based on knowledge (which, for the purposes of these discussions, includes wisdom and values)
- Learning by discovery
- Communicating with other intelligent entities (exchanging data and/or knowledge)

Let us consider each of these aspects in turn.

One of the key features of intelligent systems is that they are knowledge-driven rather than simply reacting to data in a prescribed way. When faced with uncertainty intelligent systems consult internal and/or external knowledge and decide how to react. They have a choice of behaviours and a mechanism for selecting the most appropriate behaviour (a decision mechanism). Alternative behaviours may be stored in a knowledge base as patterns (possibly in the form of scripts, algorithms, rules). Selecting behaviour amounts to selecting a pattern that appears to be the most appropriate under circumstances. If the selected behaviour proves to be inadequate, an intelligent system will re-visit its knowledge base and select a different pattern. This method is known as Trial-and-Error.

Learning by discovery means (1) analysing data on the quality of problem solving results (either own or of others) and the trail of decisions that led to these results and (2) discovering which patterns of behaviour, under which circumstances, lead to success. This process is known as Pattern Recognition.

Communication is the key activity of a distributed intelligent system – a system consisting of decision making elements (agents) connected in a network. It is a means of generating emergent properties such as emergent intelligence.

Here is a simple example of an intelligent system. A production scheduler is called intelligent if it is capable of rapidly rescheduling a production plant whenever an order is unexpectedly changed or a production resource fails. More over, to be described as intelligent a scheduler must be able to self-reconfigure and continue functioning even if some of its software components fail. Such a scheduler achieves its main goal (uninterrupted production) under conditions of uncertainty (unpredictable change in orders or software failures). An unintelligent scheduler has no mechanism for handling unpredictable events. It usually stops working (stalls) or re-start the whole scheduling process from the beginning.

Intelligent systems can be centralised or distributed. Centralised Intelligent Systems make all decisions at a single node. These systems have one hub that is responsible for all choices of behaviour. The structure of information flows to and from the decision-making hub is hierarchical. In contrast distributed Intelligent Systems are systems with decision-making elements distributed throughout the system forming nodes of a network.

There are many advantages of distributed intelligence, the main being the phenomenon of emergence. When many intelligent elements interact, the resultant behaviour is far more powerful than the sum of behaviours produced by the same elements in isolation. Thus, we are able to obtain a very intelligent behaviour from a large number of rather primitive building blocks by organising them into a network and providing them with facilities to communicate with each other. For example, collective intelligence of a swarm of bees – Swarm Intelligence – is considerably higher than the intelligence of each constituent bee. Similarly, Emergent Intelligence of a team of human decision makers is considerably higher than individual intelligence of team members.

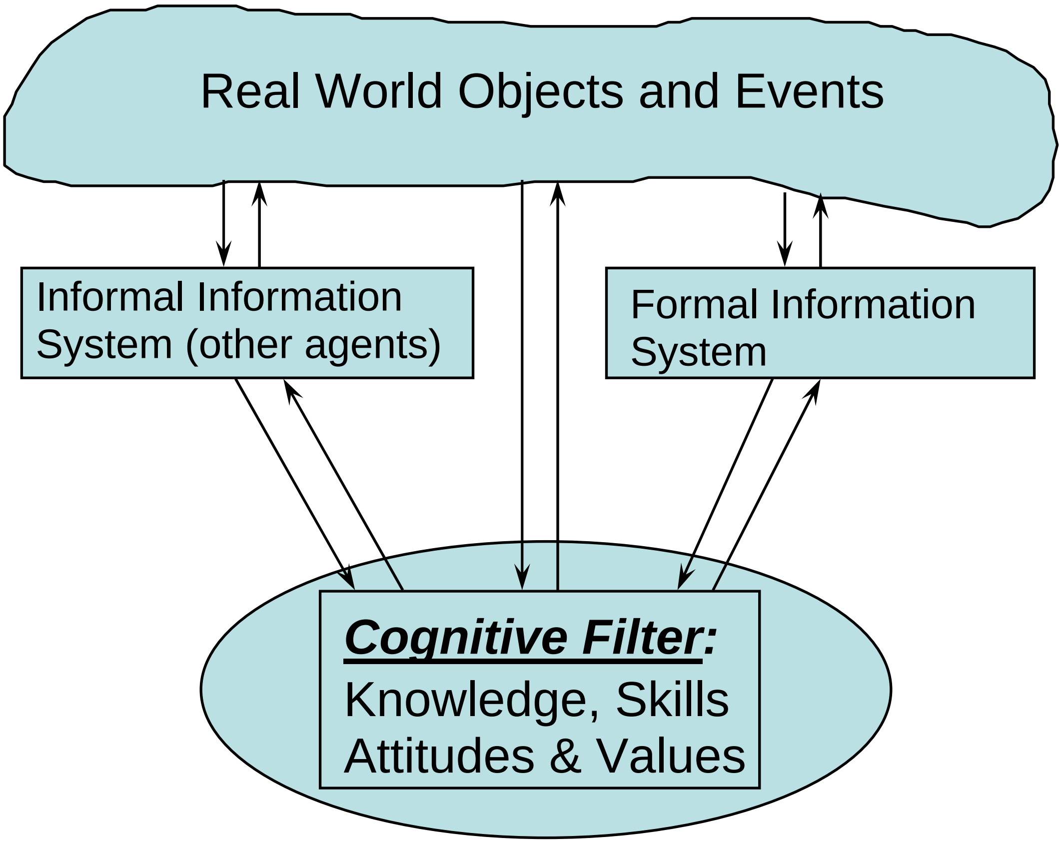
To obtain benefits of emergence it is sufficient to distribute decision elements logically. In other words, as long as decision elements are interlinked into a network they will generate emergent behaviour even if they are placed together in the same physical location (eg, in the same room or on the same server). Further benefits can be obtained, however, by geographical distribution, that is, by placing decision elements as close as practical to sources of input data (eg, on each departmental server or, even better, on each workstation in an organisation). Due to emergence and due to the closeness of its decision elements to input data sources, distributed intelligent systems are far more responsive to unpredictable changes than centralised systems. They are also amenable to scaling. For solving large problems we can build networks of decision-making networks. It is envisaged that the whole Internet will evolve into a global distributed intelligent system – a semantic web.

Emergent Intelligence

Human intelligence with all its capacities for problem solving, learning, discovery, inventions, creation of art, emotions, and even self-awareness, is a product of the interaction of billions of neurons in our brains – an emergent property of a complex system, which is a product of human evolution.

To build an artificial brain, say in software, with the same number of constituent elements is simply not practical. The alternative is to make each constituent element somewhat more intelligent (like a software agent) and then connect them into a network. The idea came to me from observations of human teamwork, where only two members of the team can produce noticeable emergent intelligence.

The subject of emergent intelligence is covered in Further Readings.



A human agent in interaction with real world objects and events

Solving Complex Problems

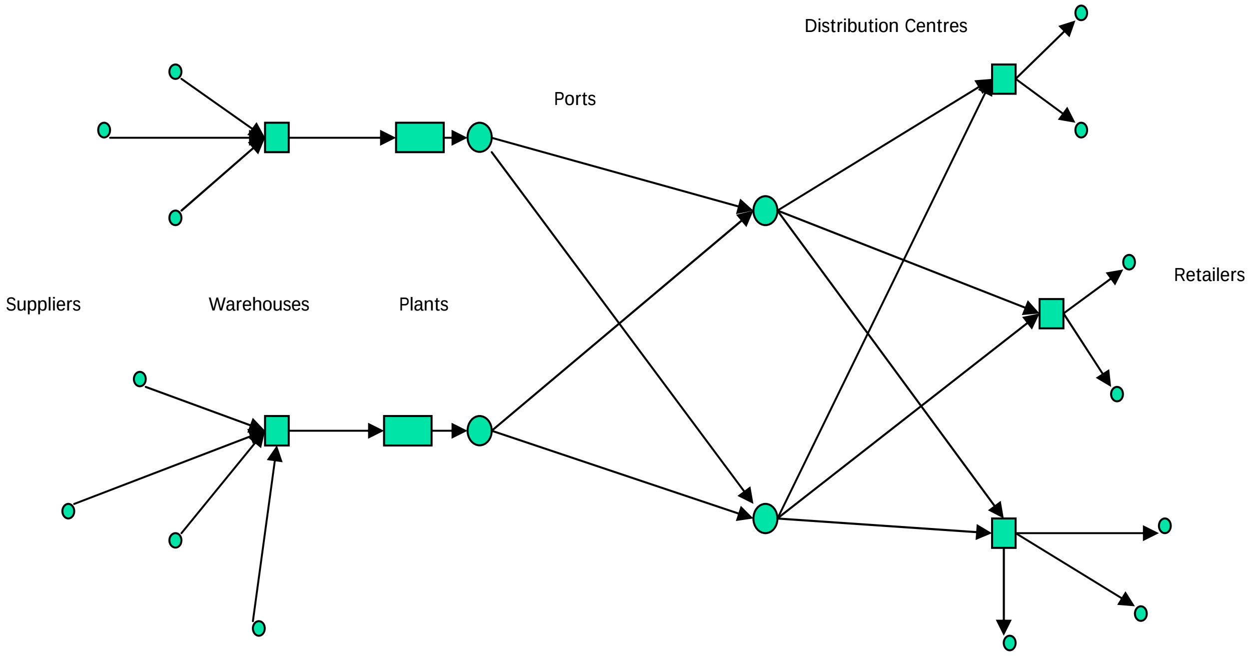
A typical problem of interest is the Resource Allocation Problem, which can be formulated as follows: Given a set of resources and a set of demands with associated attributes, find a distribution of resources to demands that will maximise/minimise a given value.

The allocation problem is complex if the *quantity* of resources and demands is large, if the *variety* of resources and demands is high, and if the *uncertainty* is present (problem is subject to unpredictable changes).

Examples include the allocation of

- Investments to projects (portfolio management)
- Budgets to business units (financial management)
- Services to orders (service scheduling)
- Goods to customers (distribution logistics)
- Parts to machine-tools (production scheduling)
- Supplies to demands (eCommerce)
- Jobs to staff (job allocation)
- Staff to projects (project management)
- Infections to subjects (epidemics containment)
- Activities to suspects (security, fraud detection)
- Information items to search queries (semantic search)
- Data records to clusters (knowledge discovery)
- Meanings to words (text understanding)

Very often these allocation problems are interconnected into vast global networks. A good example is oil production, transportation by tankers, road vehicles and pipelines and distribution to petrol stations and commercial consumers.



A global supply, transportation and distribution network

Here is an effective method for solving complex problems:

A real-life complex problem, such as an allocation problem, is modelled in software as a Virtual World in which Software Agents, each representing a real or abstract object of the problem domain, through a process of negotiation arrive at a solution. As the dynamics of the problem takes its course and frequent unpredictable real-life events keep occurring, the solution, worked out by agents, changes and adapts to each event, in real time.

For example, in a road transportation problem, agents representing orders, cargos, parcels, trucks, drivers, roads, warehouses, etc negotiate with each others how to allocate transportation resources to orders with a view to maximising profit, reducing costs or improving levels of service. Whenever a new order arrives, a truck gets delayed, a road gets jammed, etc, agents re-negotiate previously agreed schedule to accommodate the new event as best as possible, in real time. In a complex highly dynamic transportation situation the schedule is perpetually changing.

There are several types of problem solving activities associated with the Resource Allocation Problem:

- ***What-if? Analysis*** – building a model of the situation and simulating various resource and demand configurations in order to answer what-if? questions
- ***Design*** – building a model of a new resource allocation unit and running simulations in order to test the proposed model under various resource/demand configurations
- ***Planning*** – building a model of a particular resource and demand configuration, assuming certain changes, and running a simulation to determine the likely schedule
- ***Real Time Scheduling Decision Support*** – building a model of a concrete resource allocation configuration and constructing the schedule of activities taking into account every event that affects the schedule, as it happens, perpetually searching for feasible modifications to the current schedule and offering to the decision maker options as well as an advice on the best option
- ***Autonomous Real-Time Scheduling*** – as above without waiting for user decisions; the system makes scheduling decisions autonomously and communicates instructions to resources directly via an appropriate digital network

Multi-Agent Technology

Distributed intelligent systems can be effectively implemented using Multi-Agent Technology. What follows is a description of a particular brand of multi-agent technology, based on my research. Other researchers and developers have pursued other approaches and readers are encouraged to make comparisons.

Intelligent Agents (also called Smart Agents or Software Agents) are computer programs that are capable of accomplishing their goals under conditions of uncertainty through the interaction with other intelligent agents or humans.

An agent is created when needed and it achieves its goals by:

- Analysing its current task
- Deducing, from the available knowledge, how the task can be solved
- Composing messages (for other agents or humans)
- Sending messages to selected correspondents
- Receiving messages (from other agents or humans)
- Interpreting received messages
- Deciding how to react to received messages
- Implementing decisions

When an agent accomplishes its mission it is destroyed.

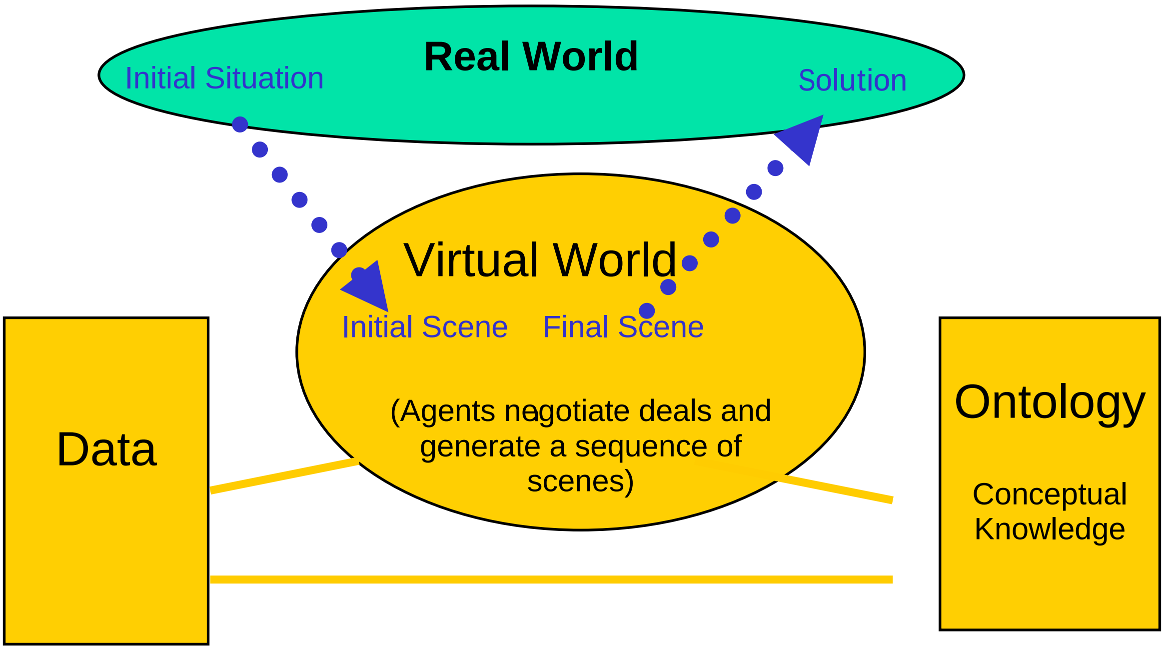
Multi-Agent Systems (also called Swarms of Agents or Societies of Agents) are systems capable of achieving their goals through the interaction of constituent agents.

Multi-agent systems solve problems in a stepwise fashion. Constituent agents negotiate among themselves how to make every step towards the goal. Note that performance of these systems depends primarily on the effectiveness of the interaction among constituent agents rather then on agents themselves. Therefore, systems consisting of a very large number of very simple agents engaged in a rich interaction with each other are particularly powerful and cost-effective. Such systems are called High Granularity Multi-Agent Systems. The comparison is often made with a colony of ants or a swarm of bees: each bee has only a limited intelligence and yet the swarm is capable of a complex behaviour resulting from the ability of bees to pass to each other useful information.

In high granularity multi-agent systems each agent is given relatively simple tasks to perform. Typical tasks given to agents are acquiring or placing resources on behalf of their clients and owners. An agent is assigned to each Client who creates Demands, to each Demand created by Clients, to each Resource available to satisfy Demands and to each Resource Owner. These Client Agents, Demand Agents, Resource Agents and Resource Owner Agents form a Virtual World in which they interact. Virtual Worlds are models of domains of the real world. Examples of Virtual Worlds include:

- Virtual Commerce, where agents allocate available goods or services to demands for goods/services, to accomplish electronic trading
- Virtual Manufacturing, where agents allocate parts and machine tools to orders, to schedule production
- Virtual Transportation, where agents allocate pallets to transportation slots, to schedule the delivery of goods to customers
- Virtual Timetabling, where agents allocate classrooms, lecturers and students to classes, to solve logistics of an educational process
- Virtual Knowledge Discovery, where agents allocate data to clusters, to obtain meaningful patterns
- Virtual Text Understanding, where agents allocate meanings to words and sentences, to determine if a text is of interest to their clients

The list of Virtual Worlds in which multi-agent systems can effectively solve resource allocation problems is very large - as large as a list of domains of the real world.



As the dynamics of the problem takes its course agents perpetually modify the scene

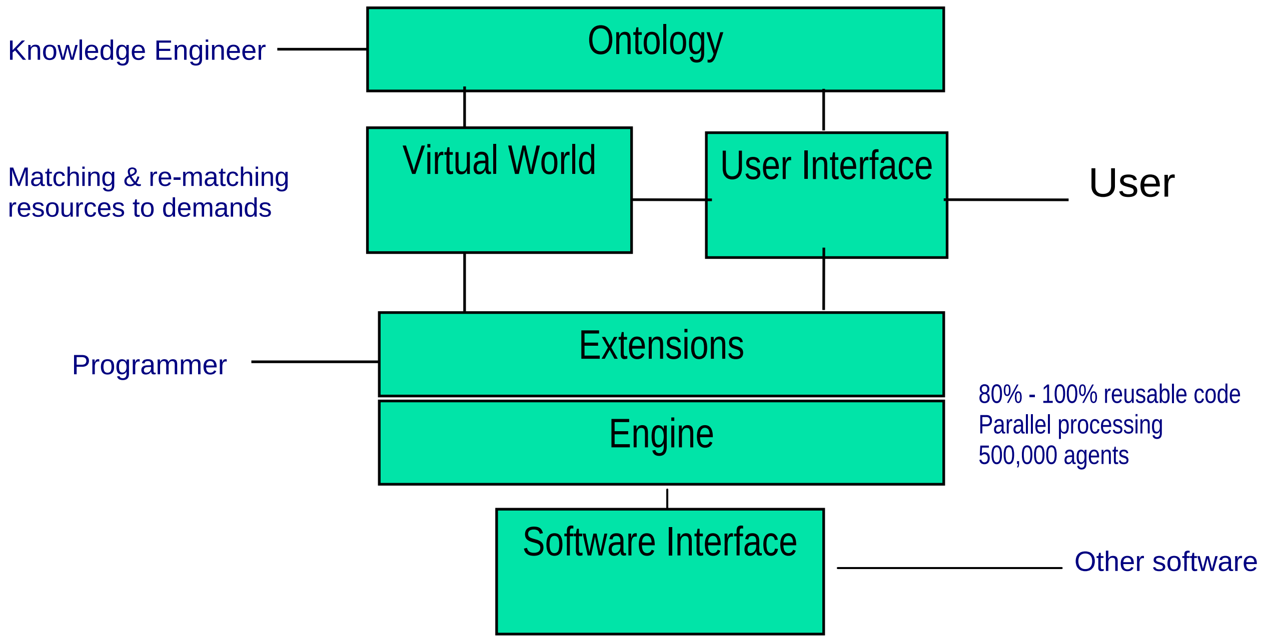
The allocation of resources within a Virtual World can be designed to work as a Virtual Market:

- *Resource Owner Agents* greet resource owners, help them to register or log in, collect data on their preferences and discover and store any knowledge that may be useful for satisfying their expectations
- *Client Agents* greet clients, help them to register or log in, collect data on client preferences and discover and store any knowledge that may be useful for satisfying their expectations
- *Demand Agents* send messages to *Resource Agents* describing their requirements and stating the amount of notional monetary units (MUs) they are prepared to pay for a resource
- *Resource Agents* send messages to *Demand Agents* describing what is on offer and how much the resources costs
- *Demand Agents* and *Resource Agents* negotiate offers and make deals
- In cases where a full demand – resource matching is not possible to achieve, agents may accept a partial matching, possibly consulting clients and/or resource owners
- The payment for a partial matching is less than the payment for a full matching
- When a new demand or resource arrives to the market, agents will try to re-negotiate a previously agreed partial matching, offering a compensation to the agent that agrees to release its purchase to another bidder
- A deal is agreed only if it increases the total value of transactions
- In a stable situation agents negotiate and re-negotiate deals until the optimal distribution of resources to demands is achieved (until they reach the point when further increase of the total value of transactions is not possible)
- In a situation characterised by a frequent change in demand/supply conditions, agents are capable of achieving a satisfactory allocation, which may or may not be optimal.
- Agents are able to switch between competition and collaboration whenever they perceive that such a change would improve their performance.

The concept of Virtual Market enables cost-effective implementation of multi-agent systems because it reduces the variety of agents and standardises their interaction in terms of trading.

An Architecture for Multi-Agent Systems

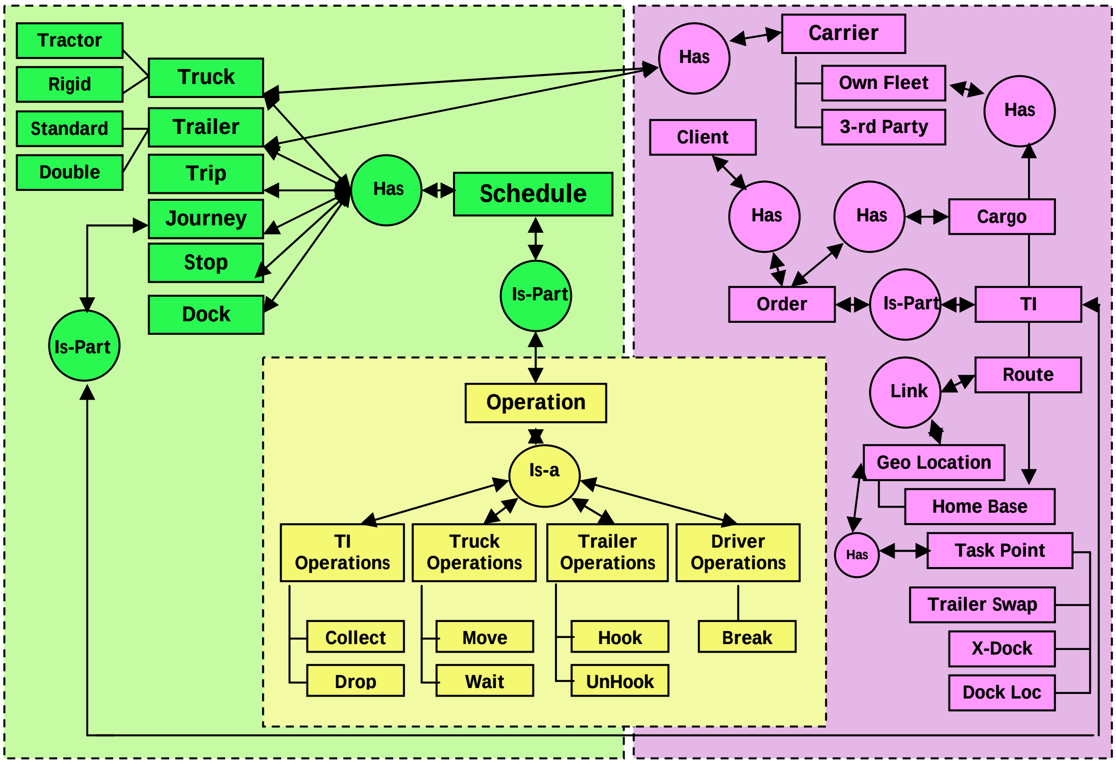
One way of partitioning multi-agent systems into key components and interconnecting them is shown below. The key components are: Ontology, Virtual World, Multi-Agent Engine, Human-Computer Interface and Software Interface, which are linked as shown below.



An Architecture of a Multi-Agent System

Ontology

Ontology contains extensive knowledge on the domain in which the system operates. The knowledge is structured in terms of a network of Objects, Attributes, Relations and Processes. The performance of agents critically depends upon the quality of the domain knowledge stored in Ontology. Application developers are given sophisticated tools for constructing application ontology, which drastically reduces order-to-delivery lead times. The same tools are given to users to modify ontology during system operation and thus adjust the system behaviour when required.

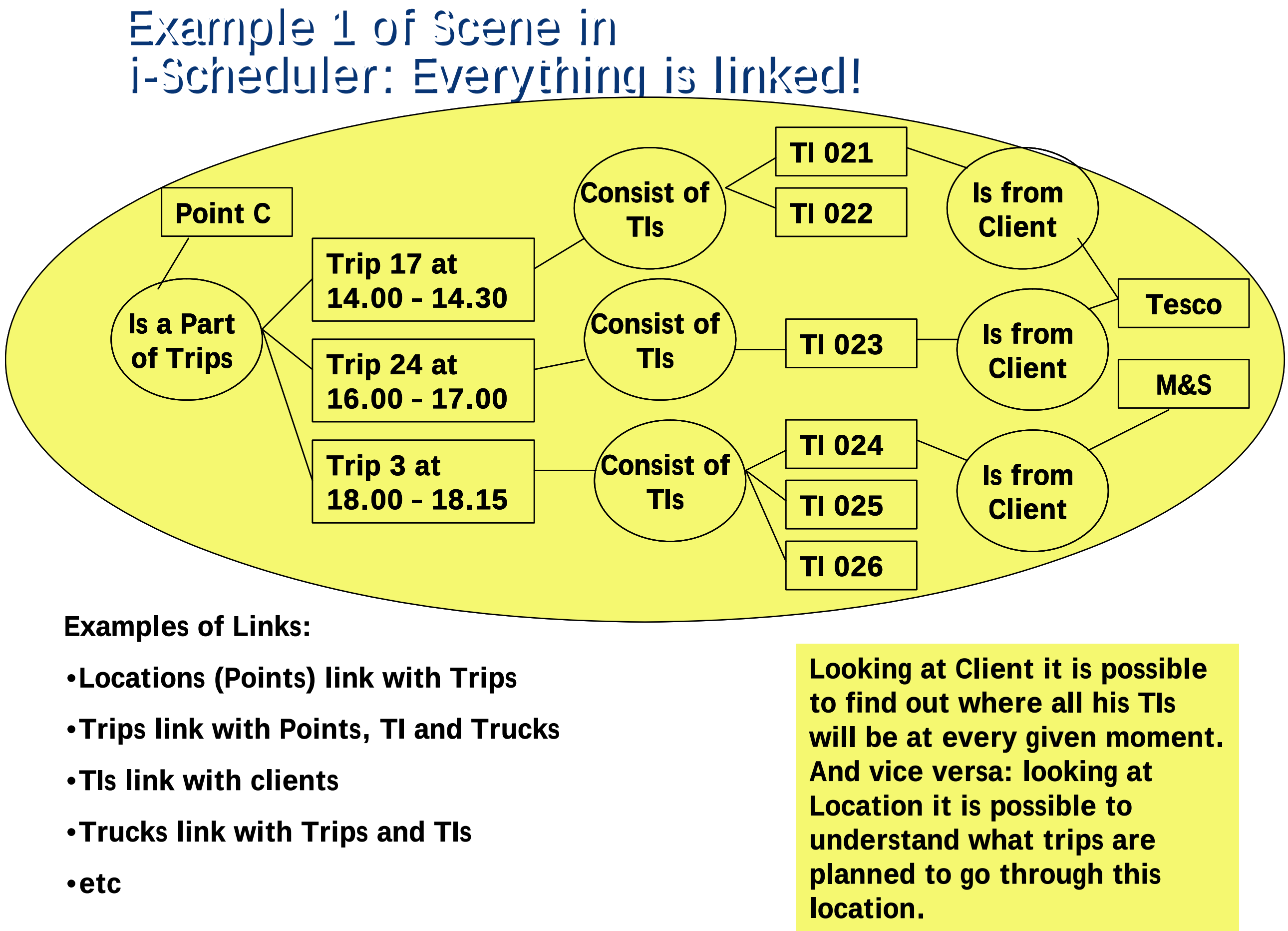


Ontology as a network of objects, attributes, processes and relations

Virtual World

The Virtual World is the place where agents are created when needed, where they solve given tasks by means of sending messages to each other, and are destroyed when their useful life comes to an end. The very large number of agents working in parallel increases the speed of resource allocation and enables scheduling and re-scheduling to be accomplished in real time.

Agents collectively work on the object known as a Scene, which represents the current real-life situation as a virtual network.



A scene for i-Scheduler: everything is linked

Multi-Agent Engine

The Multi-Agent Engine includes Runtime Algorithms with Extensions and associated Tools, that is, all the algorithms and protocols required for proper functioning of agents as well as tools for constructing ontology. The Engine is as complex as a multi-tasking operating system. It supports pseudo-parallel running of a very large number of agents and enables their interaction at great speed. Typically, 500,000 agents may be working in parallel and exchanging 50,000 messages per second. The powerful performance of a high granularity agent system comes from the Engine sitting on top of an operating system such as Windows or UNIX. Embedding agents directly into Windows or UNIX results in poor performance.

Interfaces

Interfaces link the multi-agent system with users and with other software. The interface with other software is based on international standards, including XML. A well-designed multi-agent system could be ported to any platform.

Comparing Multi-Agent Systems with Conventional Software

Conventional programs allocate resources to demands following pre-programmed algorithms in a sequential manner and therefore, when dealing with a large number of resources and demands, they require a long time to find the optimal allocation. Whenever resources or demands change, these programs start the allocation process from the beginning and if changes are frequent, they “oscillate” and cannot reach the optimal solution. Centralised intelligent systems are more flexible since they are normally driven by heuristics (rules derived from experience). Nevertheless they still solve the allocation problem sequentially and therefore cannot handle frequent changes effectively.

In contrast, high granularity multi-agent systems execute the allocation of resources in parallel. Typically, hundred of thousands of agents located on a single server or workstation work concurrently, and if the problem is distributed over many servers and workstations, the number of concurrent allocation processes can rise considerably. This explains how multi-agent systems can rapidly arrive at a near-optimal allocation of resources in real time. In cases where changes are infrequent and therefore the optimal allocation is possible, agents will systematically reconsider each concluded resource-demand matching with a view to reducing the overall cost function. This is a time consuming process, which agents will carry out in addition to any re-allocations due to changes in market conditions. Agents work solving client’s problems 24 hours a day and will continue re-negotiating partially matched deals until the best possible match is achieved or time runs out.

Agents do not have to wait for instructions. They plan and execute tasks autonomously and are capable of deciding when to compete and when to co-operate with each other. They react to any change in demand or supply without being prompted. Agents representing resources will pro-actively try to place them by searching for potential customers, offering discounts, cross-selling, making special offers and/or co-operating with other agents. Agents representing customers will actively search for resources that match their requirements and will ring their clients or send them emails when they obtain satisfactory matchings.

Typical Multi-Agent Applications

Multi-agent systems can be profitably used to solve a wide variety of problems. The domain in which agents will probably make the greatest impact is Real-Time Logistics. This is because logistics is critically affected by the dynamics and unpredictability of new turbulent markets.

1. Intelligent Logistics Systems

An intelligent logistics system is the system that allocates resources to demands in time and space and is capable of real-time execution of re-allocations whenever demands, or availability of resources, change. Systems with distributed intelligence implemented using multi-agent technology are particularly suited to problems that are complex, dynamic and where changes are frequent and unpredictable. There are many applications based on these concepts.

1.1 Supply Chain Management and Production Scheduling

A Supply Chain is the flow of goods and/or services from suppliers, via intermediaries to final assembly plants (manufacturing), display points (retail) or to end customers (order fulfilment). Production Scheduling is the flow of goods or services along production and assembly lines within a plant or office. The ultimate goal is to synchronise the scheduling of all related internal and external flows of goods and services, which may involve managing millions of physical objects and documents.

When flows of goods and services are subject to frequent and unpredictable changes, there is a need for the real-time scheduling, which can be at present achieved autonomously only by high granularity multi-agent technology. Agents are assigned to each order, flow element and each processing/storing resource and instructed to achieve optimal or at least near optimal matches of individual resources and demands.

1.2 Transportation Logistics

Following principles outlined above, multi-agent systems can schedule, in real time, the loading, unloading and movement of trucks, ambulances, cargo aircrafts, container ships and tankers as well as their crews, fuel supplies and maintenance.

1.3 Project Logistics

Whilst in the past the key factors in project management were thorough preparations and strict adherence to deadlines, under current turbulent market conditions the important feature is the adaptability and ability to handle frequent changes. The real-time scheduling and re-scheduling of projects can be effectively executed by parallel negotiations between agents assigned to human and other resources and agents assigned to tasks of fixed duration (demands) within given time slots.

1.4 Document Flow Management

The real-time allocation of documents to document processing and storing resources (human or computer-based) can be done by making an agent responsible for each document and for each document processor/store, and letting them negotiate the best obtainable matchings. Thus document flows could be managed in the same way as flows of components, subassemblies and products in an agent-based production scheduling system.

1.5 Other Logistics Applications

In addition to the most obvious domains of application, such as manufacturing, construction industry, retail, transportation, production of software, shipping etc, Intelligent Logistics Systems are applicable to many other domains including Health (real-time allocation of patients to medical staff, laboratories, operating theatres); Social Services (allocation of benefits); Law (allocation of citizens to juries) and Media (allocation of advertising slots to clients, scheduling of studio resources, personalising news delivery), to mention only few. One of the most exciting prospects is to develop operating systems for networks, desktops, notebooks and handheld computers, which incorporate agents negotiating the allocation of resources to tasks. The other “killer application” is to employ agents to design artefacts and products in a Virtual Design Studio environment.

2. Intelligent Supply-Demand Matching Systems

An intelligent supply-demand matching system is an Internet-based system that provides services, such as trading, booking, voting, participating in competitions, advertising, or searching for information, to a group of businesses and/or a community of individual users. As its name implies, the system matches available supplies to demands. These systems are simpler than logistics because the matching does not involve particular timings and locations. The following types of intelligent supply-demand matching systems are of particular interest:

- Intelligent eCommerce Systems
- Intelligent eGovernance Systems
- Intelligent eCommunity Systems
- Intelligent eLearning Systems

2.1 Intelligent eCommerce Systems

An intelligent e-commerce system is the system that executes electronic trading between businesses (B2B), or between businesses and consumers (B2C), by matching available supplies to demands. Intelligent e-commerce systems are implemented by building multi-agent technology into trading portals. Agents are assigned to supplies and demands and given tasks of searching for the best possible matching. An agent-based e-commerce system is capable of

- Achieving a partial matching of supplies to demands, when a full matching is not possible,
- Executing re-matching of a previously agreed partial matching, when supply/demand conditions change
- Autonomously discounting and cross-selling goods and services, when appropriate
- Actively pursuing matching process until the optimal match is achieved or the time runs out
- Maintaining communication contacts with customers by autonomously generating emails or phone calls.

2.2 Intelligent eGovernance Systems

An intelligent e-governance system supports an interactive relationship between central, regional or local authorities and citizens. It supports information searches related to activities of administrations, actively helps citizens to discharge their duties (enables electronic voting, electronic assessment and payments of taxes, participation in opinion polls, etc) and to ascertain their rights (to benefits, allowances, schooling etc).

2.3 Intelligent eCommunity Systems

An intelligent e-community system supports a community, that is, a group of users with common interests. Communities may be geographical (eg, all those who live in a particular area), professional (eg, members of the British Computer Society) or centred on a topic, institution or personality. Multi-agent technology contributes to the search effectiveness and user-friendliness.

2.4 Intelligent eLearning Systems
An intelligent e-learning system provides facilities for developing and delivering Internet-based educational courses. Agents help to reduce the effort and costs of course development and provide help to learners.

3. Knowledge Management Systems
Knowledge management is concerned with discovering, storing, refining and distributing knowledge. The following knowledge management systems are of particular interest:

- One-to-One Dialogue Systems
- Text Understanding Systems
- Semantic Search Engines
- Knowledge Discovery Systems

3.1 One-to-one Dialogue Systems
These are systems which include personal agents that greet website visitors and conduct conversations with them with a view to ascertaining their needs and expectations. They also contain knowledge discovery agents that accumulate knowledge about each regular visitor through a process of data clustering.

3.2 Text Understanding Systems
By assigning agents to words in a sentence and giving them a task to discover the meaning of the sentence from the context, it is possible to improve the effectiveness of search engines ensuring that they return only relevant searches. For example, it has been recently developed a system for the search for relevant scientific abstracts for subject areas where the daily output of research results is too large for human analysis.

3.3 Semantic Search Engines
Multi-agent systems that understand text can be used as search engines that match closely supplied information with search queries based on meaning rather than on key words and indexing.

3.4 Knowledge Discovery Systems
A knowledge discovery system searches for useful patterns in data and text. In a typical application, agents are assigned to records and given a task to search for similar records with a view to forming clusters. Agent technology is particularly suitable for dynamic clustering under conditions of frequent arrival of new data (as it is a case in portal applications). Whenever a new record becomes available previously established clusters are reconsidered with a view to incorporating new data into the overall clustering scheme.

Further Developments

There is a need to develop commercial multi-agent products for potentially highly lucrative agent-based systems and applications. Only three new developments will be discussed here: multi-agent peer-to-peer networks, multi-agent networks of artefacts and multi-agent person-to-person communication systems.

1 Multi-Agent Peer-to-Peer Systems
A peer-to-peer multi-agent network enables swarms of agents that reside on various servers or workstations to communicate with each other and share their computational loads. Such architecture would enable cost-effective enterprise resource planning solutions that are powerful and at the same time flexible enough to cope with complexity, dynamics and uncertainty inherent in global economy.

2 Multi-Agent Networks of Electronically Tagged Artefacts
With rapidly decreasing costs of electronic tagging it is now time to design intelligent networks, which would enable co-operative working of artefacts. I have in mind

- Household networks of tagged appliances and processors
- Manufacturing networks of tagged components, pallets, products, machines and conveyors
- Transportation networks of tagged parcels, transportation slots and store positions
- Retail networks of tagged goods, shelves, trolleys, storage positions, and transportation slots

The co-operation of tagged nodes of such networks would be negotiated among agents assigned to the nodes.

3 Multi-Agent Person-to-Person Communication Systems
SmartCard equipped members of a team passing near each other (or near SmartCard-enabled artefacts) would connect to a temporary wireless communication network, which would enable transmission of information from one personal agent to the other with a view to enhancing collaborative effort. Such systems will be most probably used for maintenance crews (eg, aircraft maintenance, medical equipment maintenance) as well for security systems.

Conclusions

As we switch from industrial to global information economy it is less desirable to continue constructing software as large hierarchies of processes exhibiting deterministic behaviour. A new paradigm is in the making, the paradigm of multi-agent software technology, incorporating large numbers of small intelligent programs concurrently interacting among themselves, making decisions based on knowledge, co-operating or competing with each other, negotiating with interested parties, matching resources to demands and reconsidering previously agreed deals whenever forced by unpredictable events in a dynamic environment.

Further Reading

A list of recommended books, papers and websites is enclosed as an appendix.

These Lecture Notes were revised September 2006.