Supply Chain Management and Multiagent Systems: An Overview

Thierry Moyaux[1], Brahim Chaib-draa[1], and Sophie D'Amours[2]

[1] Universit Laval, Dpt. d'Informatique et de Gnie Logiciel, DAMAS & FOR@C
Ville de Qubec G1K 7P4 (Qubec, Canada), {moyaux, chaib}@iad.ift.ulaval.ca,
[2] Universit Laval, Dpt. de Gnie Mcanique, FOR@C & CENTOR
Ville de Qubec G1K 7P4 (Qubec, Canada), sophie.damours@gmc.ulaval.ca,

Abstract. This chapter introduces the topic of this book by presenting the fields of supply chain management, multiagent systems, and the merger of these two fields into multiagent-based supply chain management. More precisely, the problems encountered in supply chains and the techniques to address these problems are first presented. Multiagent systems are next broadly presented, before focusing on how agents can contribute to solving problems in supply chains.

1 Introduction

This chapter presents how multiagent systems are used to manage supply chains. To this end, supply chain management is first introduced as a business practice for solving some industrial problems by providing the concept of inter-company collaboration. Such a collaboration allows planning and synchronizing operations within a supply chain in order to solve the considered industrial problems, that is, within a network of firms producing and distributing products or services to end-customers. Instrial issues and some collaboration techniques are presented in Section 2.

The second area of interest in this book, namely, multiagent systems, is next introduced. The concept of "agent" is first defined, and next compared with another concept from Computer Science, the concept of "object". After that, the general agent architectures outline the different levels of agent sophistication. Then, we motivate the use of multiagent systems, and we compare these systems with some other scientific approaches. Finally, we illustrate this section with some examples involving multiagent systems in different fields or applications. This presentation of multiagent systems is developed in Section 3.

A synthesis of supply chain management and multiagent systems extends the previous illustrations of agents applied in different fields. For that purpose, agents are first introduced as a new information technology for supply chain management. The arguments pro agents outlined in Section 3 are next extended for the special case of agents in supply chains. Some projects applying agents to supply chains eventually illustrate this section. This synthesis of supply chain management and multiagent systems is detailed in Section 4.

2 Supply Chain Management

We now introduce the concept of supply chain as a solution to some industrial issues. Specifically, industrial issues are first outlined next ,one of them called the bullwhip effect is detailed. Then, the concept of supply chain management is presented as a solution to these problems, as well as the collaboration implied by this concept. Finally, information technologies are presented as a tool supporting collaboration.

2.1 Industrial Problems in General

First, companies face a huge number of problems, such as how to make decisions concerning production planning, inventory management and vehicle routing. These three decisions are managed separately in most organizations because making each individual decision is very difficult, since many constraints have to be satisfied (production, shipping and inventory capacities, precedence order of activities, legal obligations, etc.) [1]. For instance, the multistage, multicommodity inventory management problem and the vehicle routing problem are both known to be NP-hard problems ([2] cited by [1]), i.e., very difficult.

Secondly, the problem is yet harder in reality because the decisions concerning production planning, inventory management and vehicle routing are *interdependent*. Hence, these three decisions should be taken together, which makes the planning problem harder.

Third, companies are not isolated, but impact on and are impacted by their partners. As a result, when a company maximizes its profits, it may disturb other companies, which may result in globally underoptimal decisions, because organizations may have different conflicting objectives [3, pp. 3]. The best solution would be to make the decisions together concerning production planning, inventory management and vehicle routing for several companies. As this planning problem is hard for a single company, synchronizing all companies decisions together is very hard.

The concept of supply chains was proposed to address this problem of minimization of total supply chain cost, while meeting fixed and given demand by points-of-sale, e.g. by retailers [4, pp. 8]. Before presenting supply chains, we focus on one particular example of industrial issues: the bullwhip effect.

2.2 A Particular Example of Industrial Problems: The Bullwhip Effect

We now present a phenomenon occurring in supply chains called the bullwhip effect. This effect consists in an amplification of the order variability. This variability is a problem because it makes demand (i.e., orders) more unpredictable. Figure 1 shows how this effect propagates in a simple supply chain with only three companies: a retailer, a wholesaler and a paper mill. In this figure, the retailer exclusively sells to the customer and buys from the wholesaler, the wholesaler sells to the retailer and buys from the paper mill, and the paper mill sells to

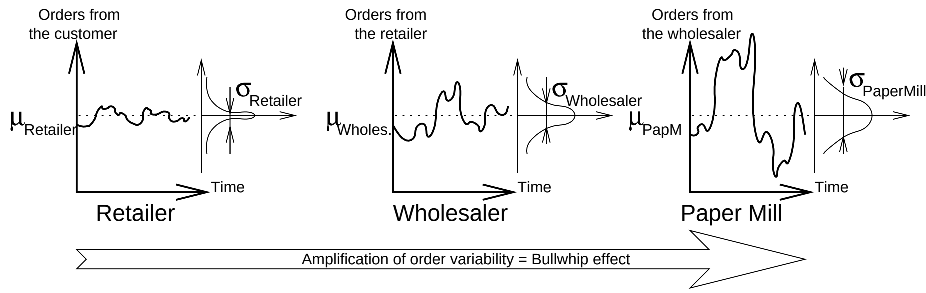

Fig. 1. The bullwhip effect [5, 6].

the retailer and buys from an unknown supplier. The ordering patterns of the three companies are similar in the way that the variabilities of an upstream site are always greater than those of the downstream site [5]. As a variability, the bullwhip effect is measured by the standard deviation σ of orders. Note that the means μ of orders are all equal in our example given in Figure 1.

There are several consequences of the bullwhip effect. It has been estimated that these consequences altogether would increase the costs by 200-300 MFIM (33-50 million euros) annually for a 300 kton paper mill [7] in Northern Europe. The consequences causing such costs are as follows:

1. *Higher inventory levels*: Every participant in the supply chain has to stockpile because of a high degree of *demand uncertainties and variabilities* induced by the bullwhip effect [6];
2. *Supply chain agility*[3] *reduction*: As inventory levels are high (cf. previous consequence "higher inventory levels"), the supply chain should sell products in inventory, before it sells the new products demanded by end-customers, which generates inertia in following end-customer demand. Moreover, demand uncertainties induced by the bullwhip effect make it more difficult for the supply chain to understand which product is demanded by end-customers;
3. *Decrease of customer service levels*: Demand variabilities may incur stock-outs, in which case, no products are available to be sold, and thus, no service can be given to customers;

The last two consequences of the bullwhip effect are related to the difficulties of planning under uncertainties, and consequently lead to:

4. *Ineffective transportation*: Transportation planification is made more difficult by demand uncertainties induced by the bullwhip effect;
5. *Missed production schedules*: Similarly to transportation, production planification is made more difficult by demand uncertainties induced by the bullwhip effect.

[3] The Iaccoca Institute [8] (cited by [9]) defines agility as the "ability of an organisation to thrive in a constantly changing, unpredictable business environment".

Several causes have been proposed to explain the appearance of the bullwhip effect, such as demand signal processing which uses forecasting methods not perfectly accurate, gaming among companies when demand exceeds supply, order batching which discretizes orders, and price variations which incite clients to over-order when price is low [6]. However, such a sharing of information is only possible when companies collaborate, because demand information is very important and should normally be kept secret Sharing demand information is often said to be the solution to the bullwhip effect (or at least a part of the solution) [3].

Notice that an interesting feature of the bullwhip effect is that it creates a link between the companies, which makes the concept of supply chain interesting. To see this link, consider the retailer in Figure 1 which does not suffer directly from the bullwhip effect, because it receives a quite constant demand from end-customers. As a consequence, the retailer has no direct incentives to reduce this phenomenon, while this would benefit the rest of the supply chain, and thus, the retailer itself indirectly because products may be cheaper, more available, etc.

We shall detail a thorough description of the bullwhip effect, a solution to it, and the incentives companies may have to use such a solution in Chapter 10. But at the moment, we present the concept of supply chain as a solution proposed for the industrial problems outlined in Subsections 2.1 and 2.2.

2.3 The Concept of Supply Chains as a Solution

We have not found the first definition of the term "supply chain", but we have found, for example, that Burns and Sivazlian [10] referred to it in the late 1970's. According to Muckstadt and his colleagues [11], there are many definitions and interpretations of the term "supply chain management". These authors defined a supply chain as "the set of firms acting to design, engineer, market, manufacture, and distribute products and services to end-consumers". In general, this set of firms is structured as a network, as illustrated in Figure 2 [12, 13] in which we can see a supply chain with five levels (raw material suppliers, tier suppliers, manufacturers, distribution centers and retailers). In the same context, Shapiro [4] noted that "supply chain management is a relatively new term that crystallizes concepts about integrated business planning that have been espoused by logistics experts, strategists, and operations research practitioners as far back as the 1950s". Similarly, Simchi-levi and his colleagues [3] defined this term as "a set of approaches utilized to efficiently integrate suppliers, manufacturers, warehouses, and stores, so that merchandise is produced and distributed in the right quantities, to the right locations, and at the right time, in order to minimize systemwide costs, while satisfying service level requirements". Poirier and Reiter [14] noted that the concept of supply chains improves the competitive position of collaborating companies, because it supports the creation of *synergies* among these companies. In particular, such synergies are due to the fact that a supply chain is a system, and as a consequence, this system is superior to the sum of the constituting companies. As previously explained, the concept of inter-company *collaboration* is a way to create such synergies in a supply chain.

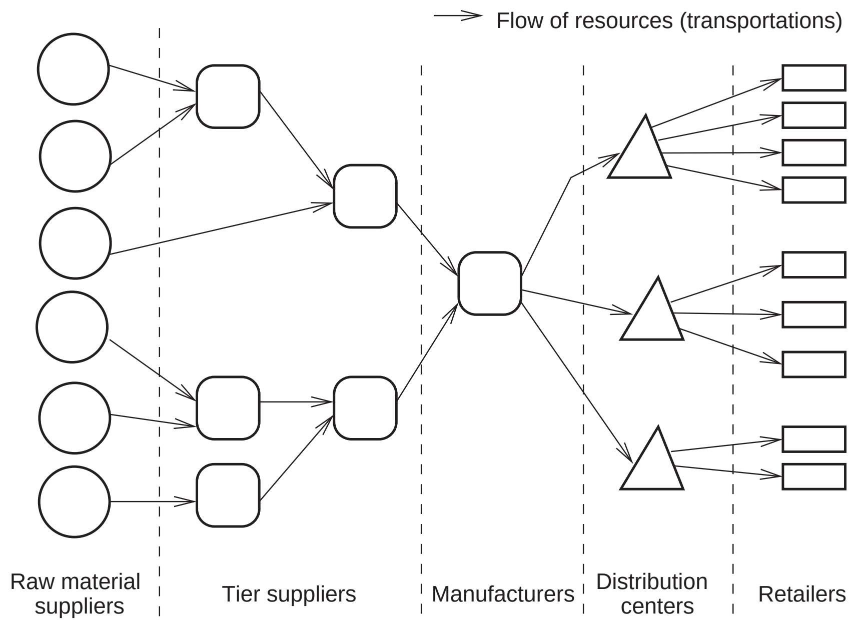

Fig. 2. An example of supply chain [12, 13].

2.4 Collaboration in Supply Chains

Muckstadt and his colleagues [11] noticed that the term *collaboration* is confusing because it has taken on several interpretations when used in the context of supply chain management. For example, various levels of collaboration techniques based on *information sharing* were set up in real supply chains. It is important to note that we refer in this chapter to collaboration as information sharing, even if collaboration is in general wider than only information sharing. We represent in Figure 3 how some of these information sharing techniques overlap. These techniques are essentially information centralization, Vendor Managed Inventory/Continuous Replenishement Program, and Collaborative Planning Forecasting and Replenishment. They are now reviewed in detail:

– *Information centralization:* This is the most basic technique of information sharing in which retailers broadcast the market consumption (approximated as their sales) to the rest of the supply chain. As we also refer to information centralization, it is necessary to distinguish information sharing from information centralization: the latter is a particular case of the former, because information centralization is the multi-casting in *real-time* and *instantaneously* of the market consumption information, while information sharing is only the sharing of the demand information between any companies. Moreover, several kinds of information may be shared, such as their available production capacity, their inventory level... and from this viewpoint, information sharing includes information centralization.

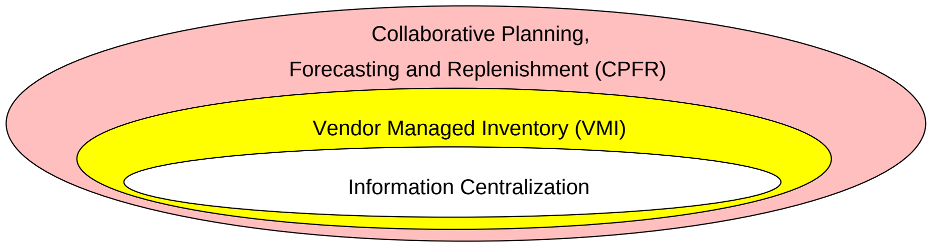

Fig. 3. Overlap of some collaboration techniques through information sharing.

In the context of the bullwhip effect, Chen and his colleagues [15] formally showed, for two forecasting methods, that information centralization reduces this effect. In fact, information centralization reduces it, because each level of the chain can base its forecasts on the actual market consumption, instead of basing them on incoming orders, which can be much more variable than the actual market consumption [5, 6].

– *Vendor Managed Inventory (VMI)* [16] and *Continuous Replenishment Program (CRP):* These two collaboration techniques are very similar, but are used in different industries. The idea is that retailers do not need to place orders because wholesalers use information centralization to decide when to replenish them. Although these techniques could be extended to a whole supply chain, current implementations only work between two business partners. In fact, many customers are attracted to these techniques, because they mitigate uncertainty of demand, a consequence of the bullwhip effect. Moreover, the frequency of replenishment is usually increased from monthly to weekly (or even daily), which benefits both partners. These techniques were popularized in the late 1980's by Wal-Mart [17] and Procter & Gamble [18]. In particular, VMI has became one of the key elements of the quick response program in the grocery industry [19].
– *Collaborative Planning, Forecasting and Replenishment (CPFR):* This technique developed by the VICS Association [20] (Voluntary Interindustry Commerce Standards) is a standard that enhances VMI and CRP by incorporating joint forecasting. Like VMI and CRP, current implementations of CPFR only include two levels of a supply chain, i.e., retailers and their wholesalers. With CPFR, companies electronically exchange a series of written comments and supporting data, which include past sales trends, scheduled promotions, and forecasts. Conversely to the previous two techniques, thus CPFR shares more information than only demand information. This allows the participants to coordinate joint forecasts by focussing on differences in forecasts. Companies try to find the cause of such differences and agree on joint, improved forecasts. They also jointly define plans to follow when specific contingencies occur [3, pp.239].

2.5 Supporting technologies

These three techniques of information sharing, i.e., information centralization, VMI/CRP and CPFR, can be supported by information technologies such as e-Hubs [21]. The basis of these information technologies is currently the Internet, but other technologies are also used, e.g. the protocol for Electronic Data Interchange (EDI). The first advantage of the Internet over every other technology is to provide a low-cost communication infrastructure available almost anywhere in the world. This first advantage allows companies to increase information streams, and more precisely in our context, to share more information. The second advantage of the Internet is to provide some standardized file formats (HTML, OWL, etc.), which reduces the cost of information technologies.

We shall develop the goals of information technologies and the way to achieve these goals during the presentation of the application of multiagent systems in supply chain management in Section 4.

3 MultiAgent Systems

We now focus on the second field addressed in this book, i.e., multiagent systems. We first define the concept of agents, next we compare this concept with another concept from software engineering, i.e., the concept of object. Then, we outline some architectures of agents and some arguments in favour of the use of agents in general. Finally, we illustrate projects involving multiagent systems in different areas.

3.1 The Concept of Agents

Intelligent agents are a new paradigm of software system development. They are used in a broad and increasing variety of applications [22–24]. For a long time, there was no single definition of an agent and a multiagent system: several definitions cohabited in the past [25]. Nowadays, it seems that researchers agree on the following definition proposed by Wooldridge and Jennings [26]:

> The term "agent" denotes a hardware or (more usually) software-based computer system, that has the following characteristics:
>
> **Autonomy:** agents operate without the direct intervention of humans or others, and has some kind of control over its actions and internal state;
>
> **Social ability:** agents interact with other agents (and possibly humans) via some kind of agent-communication language;
>
> **Reactivity:** agents perceive their environment, (which may be the physical world, a user, a collection of other agents, the Internet, or perhaps all of these combined), and respond in a timely fashion to changes that occur in it;
>
> **Pro-activeness:** agents do not simply act in response to their environment, they are able to exhibit goal-directed behaviour by taking the initiative.

3.2 Comparison with Objects

Based on this concept of agent, Shoham [27] proposed a new programming paradigm called Agent-Oriented Programming (AOP) to replace the current Object-Oriented Programming (OOP). The difference between agents and objects is sometimes missed by programmers familiar with object-oriented languages, such as C++ [28] or Java [29]. The main difference between these two concepts is the *autonomy* of agents. In fact, while objects encapsulate some state on which their methods can perform actions, and in particular the action of invoking another object's method, an object has control over its behaviour. That is, if an object is asked to perform an action, it always does so, while an agent may refuse. Concerning this point, Wooldridge [30] recalls the slogan "Objects do it for free; agents do it because they want to".

Of course, some sophisticated objects may be very similar to agents. In fact, Wooldridge [31] noted that there are clear similarities, but obvious differences also exist. Let us consider the case of objects in Java that can easily be transformed into threads exhibiting some behaviour. Such active objects have some autonomy like agents, but their behaviour is only procedural in reaction to message requests. On the other hand, autonomy of agents makes them perform activities without external intervention [32]. In short, object-based concurrent programming has some relationships with distributed artificial intelligence [33].

But objects and agents also present differences. In particular, object state is much simpler than agent state. In fact, an object state is only a data structure, i.e., an aggregation of variables of different types (integers, booleans, character strings...) in a common structure, while an agent state consists of components such as beliefs, decisions, capabilities and obligations. As an agent state is more sophisticated, it is also referred to as a mental state [27].

Finally, it is important to note that agents have been programmed in C++ or Java, i.e., with an OOP language, but AOP languages have appeared. For example, JACK™ designed by the Agent Oriented Software Group [34] (Melbourne, Australia) is an AOP language. This language implements concepts from AOP upon an OOP language. That is, JACK™ provides an AOP compiler transforming JACK™ code into Java code. Next, the JACK™ compiler calls the Java compiler to transform the generated Java code into a runnable Java bytecode that works on any Java Virtual Machine.

3.3 Agent Architectures

In the same manner that there are several languages to implement agents, there are also different levels of complexity of this implementation. Such complexity depends on the task that agents have to carry out and on the environment surrounding them. Russell and Norvig [35] propose the following classification of agent architectures:

1. *Simple reflex agents*: This type of agent is the simplest, because percepts are directly related to actions via some condition-actions rules. What has occurred in the past is ignored, because these agents have no memory.

2. *Model-based reflex agents*: As agents cannot perceive their whole environment, model-based reflex agents keep track of the part of their environment they cannot currently observe. To achieve this, they have an internal representation of their environment, called a "model of the world", to guess the evolution of the environment and the impact of the agent's actions on this environment. Like simple reflex agents, model-based reflex agents select their action according to condition-action rules, but now, the condition only depends on the model of the world, and not on the current perception of the environment.
3. *Model-based, goal-based agents*: This type of agent has goal information describing desirable situations, because the current state of the model of the world is not always enough to select an action efficiently. That is, the model of the world is used to elaborate some predictions on how the world would be if the agent executes an action and what is the price to pay for that. The action to carry out is chosen so that the goals will be satisfied according to such predictions.
4. *Utility-based agents*: Goals just differentiate wishable states from non-wishable states, without further details, such as, the speed, the price or the safety to reach a wishable state. As a result, in order to improve the quality of agent behaviour, agents can be given a utility function mapping its state (or a sequence of states) in the model of the world, onto a real number describing the associated degree of agent's happiness. In comparison with goal-based agents, utility-based agents do not decide which action to do in order to achieve a goal, but which action to do to increase utility. This difference implies that both types of agents find which actions to do to achieve their goals, but utility-based agents find the best actions according to some given metrics. This agent architecture is hence much nearer to the definition of Economics agents (that only maximize their utility) than the previous three architectures.
5. *Learning agents*: Turing [36] has noted the huge amount of work it takes to program an intelligent machine, and has concluded that it would be easier to build learning machines and then to teach them. Another advantage of learning agents is their adaptability to unknown environments, and the improvement of their behaviour with time. Learning agents use the feedback from a critic to learn which perceptions of the environment are desirable, and in consequence, how to behave. Precisely, agents' learning consists in improving their future performance based on their past feedback from the critic, by optimizing their behaviour such as to maximize their utility when the world continues evolving as it has been. This kind of learning makes agents discover that some kind of (but not exactly) condition-action rules always do the same thing, based on their current knowledge.

   A problem arises here: after some learning time, agents are always going to do the same things because of these discovered rules, though the agents are not sure that these actions are optimal, while they might have a better performance if they had a wider knowledge of their environment. In fact, they should try to do very different actions than those prescribed by their learning process. Some *exploration* of new actions should be carried out instead of only *exploiting* the learned knowledge.

3.4 Motivations for Multi-Agent Systems

Huhns and Stephens [37] noted that multiagent systems are generally less efficient than centralized solutions, because the distribution restrains optimization. But these authors also gave several advantages of multiagent systems. First, multiagent systems are easier to understand and implement, when the problem itself is distributed. This allows the multiagent system to give more flexibility when taking into account the modularity of the real, modelled system. Next, the distribution may force programmers to propose new algorithms to solve problems. In particular, the concurrency can be used to accelerate problem solving. Finally, a centralized solution may be impossible, because systems and data are in independent organizations. We develop this latter argument in Section 4, because it is the main one in favour of multiagent systems in supply chains.

Jennings [38] pointed out the flexible, high-level interactions of agents, that make the engineering of complex systems easier. This author recalls that *complex* systems are always distributed, and from his point of view, agent decomposition is very important to manage complexity. It follows from this, that designers need a means to reduce the complexity of the system control, in order to enhance their ability to model, design and build complex, distributed systems. Multiagent systems provide designers with this means through the decentralisation of control. In particular, the system complexity makes it very difficult to know every possible interaction in the system, because the system only has partial control and observability over its environment, and thus, this environment is highly unpredictable. Multiagent decentralisation takes this into account by letting each agent continuously coordinate its actions with other agents, instead of making this agent apply a behaviour prescribed at design-time. In short, some advantages of multiagent systems is the fact that modelling with agents:

- partitions the problem space of a complex system efficiently;
- is a natural way to modularise complex systems;
- focusses on the organizational relationships in complex systems.

Similarly, Wooldridge [30] says that *interaction* is now seen by most programmers as an important characteristic of complex softwares. For this reason, interactions, and thus multiagent systems, take a growing part in software engineering. Moreover, multiagent systems are an interdisciplinary field. For example, interactions in multiagent systems are also interesting to model dynamics in human societies.

We should note that there are also objections to multiagent systems. We now review such objections.

3.5 Differences between MultiAgent Systems and Other Fields

In general, objections to multiagent systems are due to their similarity with other fields. To respond to these objections, Wooldridge [30] points out the difference between this field and some others:

Distributed/concurrent system

- *Similarity*: By definition, multiagent systems are a special case of distributed/concurrent systems. Therefore, experience in this field has to be kept by the multiagent system community, in particular to avoid discovering again how to manage mutual exclusion over shared resources, how to avoid dead- and livelocks. . .
- *Differences*: First, agents are autonomous, and therefore, synchronization and coordination are not structured at design-time, as they are in distributed/concurrent systems. In fact, agent synchronization and coordination is achieved at run-time. Secondly, agents are in general self-interested, while components in a distributed/concurrent system have the common goal of maximizing the overall system efficiency. For these two reasons, negotiation is important in multiagent systems, while it is unknown in distributed/concurrent systems.

Artificial intelligence

- *Similarity*: Historically, multiagent systems were born from Distributed Artificial Intelligence, which is a subfield of Artificial Intelligence [39].
- *Differences*: First, the main topic of artificial intelligence has been the study of components of intelligence (learning, planning, understanding images. . .), while the goal of research about agents is the integration of these elements. Therefore, during agent implementation, much more time is spent with computer science and software engineering, than with artificial intelligence. Secondly, social ability in systems has been ignored by artificial intelligence, while this is as important in an intelligent behaviour as learning or planning.

Economics/Game Theory

- *Similarity*: Like multiagent systems, Economics and Game Theory also deal with self-interested agents, and more precisely with their interactions. Some well-known reseachers have contributed to both computer science and economics/game theory, such as von Neumann and Turing. However, these two fields have been dissociated since these beginnings. Now, the situation is changing because game theory has more and more applications in multiagent systems, and economists are interested in multiagent simulations to understand inter-agent interactions.
- *Differences*: First, concepts in economics/game theory are descriptive, and thus, indicate nothing about how to compute a solution. Such computing is often very hard [40]. Secondly, game theory is built on the notion of rationality, but some debates are concerned to the question of its validity and/or utility for artificial agent societies. Thirdly, Boutilier [41] proposes another difference that is also related with rationality. This difference is about the assumption in economics/game theory that agents

are rational (the research questions concern the social consequences of this hypothesis), while programming this rationality is the problem itself in multiagent systems.

Social Science

- *Similarity*: Social sciences study the dynamics of human societies, while multiagent systems are concerned with artificial societies.
- *Difference*: It is not certain that the best way of building artificial societies is to base them on human societies. Moreover, other tools, such as the aformentioned game theory, also model human societies, and may thus be applied.

Because of similarities between multiagent systems and other fields (distributed/concurrent systems, artificial intelligence, economics/game theory and social science), agents have been applied in some of these fields. Furthermore, they have also been applied in many real-world applications, that are, in general, functionnally or geographically distributed. We now present some of these applications of multiagent systems.

3.6 Some Applications of MultiAgent Systems

Multiagent systems have been used in many fields, as presented by Chaib-draa [24], Wooldridge [30] and Jennings [42], and Weiss [43]. As an illustration, we now outline some of these applications. Jennings, Sycara and Wooldridge [39] classify these applications in four classes:

Industrial applications: Industry was one of the earliest users of agent technology, especially in the following areas:

- *Manufacturing:* For example, the *Holonic Manufacturing Systems* (HMS) project [44–46] aims at standardizing architecture and technologies for open, distributed, intelligent, autonomous and cooperating systems in industry. Each component of these systems is controlled by agents, called "holons" for the combination of "holos" (the whole) and "on" (a particle) [46]. Each holon's goal is to work with the other holons, in order to control a production system in an efficient, scalable, open way. Applications of holons are, for instance, concurrent engineering, collaborative engineering design, and manufacturing enterprise integration [47].
- *Process control:* Process control is at a lower level than manufacturing, because manufacturing aims at controlling several workstations, while process control focusses on a single workstation. In fact, the complexity of a workstation may require the decomposition of its control into agents.
- *Telecommunications:* Telecommunication networks are geographically spread over a large area. Using agents to manage such networks is thus a natural metaphor. For instance, British Telecom [48] has developed the ZEUS Agent Building Toolkit for this purpose.

- *Air-traffic control* [42, 49]: OASIS is an air-traffic control system used at Sydney airport in Australia. Aircraft and the various air-traffic control systems are seen as agents. Agents are created when they approach Sydney airport. Their behaviour is both goal-directed ("I want to land"), and reactive to take real-time constraints into account. Some similar air-traffic control systems were designed for NASA [50], or by Cammarata and her colleagues [51].
- *Transportation systems:* Like telecommunication networks, the geographical distribution of transportation lead to the fact that agents are a natural metaphor. For example, *Automated Highway Systems* [52] unites several projects aiming at fully automatizing vehicle driving. Several goals are addressed, such as driving a vehicle without human intervention and collaborative driving. This second example consists of forming platoons of vehicles on roads, in order to improve the fluidity of traffic. Each vehicle is seen as an agent that tries to form a team with other vehicle-agents sharing the same part of trip.

Commercial applications: While agents for industry are quite often designed for a single, specific application depending on the company, commercial agents tend to be designed for a widespread diffusion. Among the areas of commercial agents, we can find:

- *Information management:* Since the users of Internet are more and more overloaded by information, agents can help them by filtering and gathering accurate information.
- *Electronic commerce:* Since Internet takes up a growing place in our everyday life, e-commerce promises to be more frequently used in the near future. In fact, agents can:
  - replace us to look for the products that best fit our needs;
  - bid for products on auctions sites, such as eBay [53], following a given strategy [54, 55];
  - try to form a coalition with agents buying a similar product, in order to have a price reduction due to the higher bought quantity [56–58].

  In particular, TAC (Trading Agent Competition [59]) aims at confronting agents to find the best buying strategy in situations close to real-life [60, 61]. TAC has several tracks, and one of them is about supply chain management.
- *Business process management:* Information systems are spread among the different departments in a company in order to bring information together. Using agents can make this information collection easier and more efficient. The collected information is useful for company managers when they make business decisions.

Entertainment applications: Although this industry is not seen as serious in computer science, it is currently growing. Specific areas of entertainment agents are:

- *Games:* For example, the concept of agents was applied in the game "Creatures" by Grand and Cliff [62] to build artificial pets living together in a simulated environment. These animals are built to resemble real-life animals, and in particular, their "brain" is a neural network.

- *Interactive theatre and cinema:* In these systems, users 'enter' the movie to play a role in this movie, and to interact with other characters played by artificial agents. Programming these agents so that they resemble real people is an issue, because they have to look like human beings, to behave like them.

Medical Applications: Agents are more and more used in medical applications, for instance:

- *Patient monitoring:* For instance, Hayes-Roth *et al.* [63]'s system Guardian is distributed to respect the fact that a team in a Surgical Intensive Care Unit is made up of people who have different expertise and who collaborate. Guardian has a hierarchical structure, in which a control agent controls perception/action agents and reasoning agents, in order to help manage patient care.
- *Rescue team management:* RoboCup Rescue [64, 65] is a competition involving the simulation of an earthquake similar to Kobe (Japan) in 1995. Agents model teams of firemen, policemen and ambulances, that have to be coordinated in order to minimize both the number of dead civilians and the number of destroyed buildings. The idea is that an earthquake scenario cannot be studied in real-life, and thus, has to be simulated in order to find which behaviour rescue teams should have.

Some other applications cannot be put in these four classes. For instance, *interface agents* assist users in software, like the paper clip in MS-Office [66], even though this is currently a mono-agent system. *Simulations of ecological and social* systems are another kind of application of multiagent systems. For example, Franchesquin and Espinasse [67] programmed a multiagent simulation, that takes into account both ecological and social dynamics, in order to study the hydraulic management of the Camargue (south of France).

Since we focus in this book on agents in industry, and more precisely, on supply chain management, we will describe additional multiagent systems in industry. It is worth noting here, that the HMS (holonic manufacturing) project, presented above, looks similar to multiagent systems for supply chain management, but it is indeed different. In fact, the HMS project addresses problems at a lower level, that is, intra-company, while supply chains are made up of several companies. As a consequence, assumptions about agents in supply chains (selfishness, available information, etc.) have to be a bit different than assumptions about holons. The other chapters in this book make such supply chain-related assumptions.

4 MultiAgent Systems in Supply Chain Management

The first section of this chapter introduced supply chain management, and the second one multiagent systems. We now focus on the merging of these two fields into multiagent-based supply chain management. We first show how computers are currently used in supply chains, then we give some arguments justifying the use of multiagent systems in supply chains, and finally some examples illustrate this section.

4.1 Information Technologies in Supply Chain Management

According to Simchi-Levi and his colleagues [3], "information technologies is an important enabler of effective supply chain management. Much of the current interest in supply chain management is motivated by the possibilities that are introduced by the abundance of data and the savings inherent in sophisticated analysis of these data". It follows that information technologies in supply chains pursue three goals:

– *collecting* information on each product from production to delivery or purchase point, and providing complete visibility for all parties involved;
– *accessing* any data in the system from a single-point-of-contact, e.g. from a PDA linked to the company information system through a wireless link;
– *analyzing* data, planning activities, and making trade-offs based on information from the entire supply chain.

To achieve these activities, information technologies use certain means:

– information technology *infrastructure* (network, databases...);
– *e-commerce;*
– *supply chain components*, which are the various systems directly involved in supply chain planning, i.e., Decision Support Systems (DSS).

The *standards* gathering these three means are, for example, the protocol for Electronic Data exchange (EDI). Although regarded as a success because it is used by large corporations, EDI was never accepted by the majority of the communities of the business world as a means of trading electronically, because its is a barrier for small companies [68]. This explains why new Internet-based standards currently emerge. In particular, the eXtended Markup Language (XML) [69] is used in more and more applications on the Internet. But XML is too generic to enable collaboration in supply chains. Therefore, some XML-based standards were proposed, such as the Resource Description Framework (RDF) [70] to define a common vocabulary for describing resources, the Web Ontology Language (OWL) [71] to give semantics to Web pages, the Common Business Library (CBL) [72] for describing documents such as orders or catalogues, etc. Please refer to Singh and Huhns [73]'s book for an overview of these technologies and many others.

Concretely, information and decision technologies take the form of:

– *Enterprise Resource Planning* (ERP) is a class of software systems organizing and managing companies [74], e.g., PeopleSoft/Oracle [75], or SSA Global [76];
– *E-commerce*, and in particular marketplaces, such as Commerce One [77] and Ariba [78];
– *Advanced Planning and Scheduling* (APS) is a class of software for Decision Support System (DSS) in supply chains.

According to Shapiro's decomposition of information technologies [4], the first two applications (ERP and e-commerce) belong to "Transactional Information Technologies" because they are concerned with acquiring, processing and communicating raw data. On the other hand, APS and DSS belong to "Analytical Information Technologies" because they allow analyzing raw data in order to help managers, which is a task at a higher level. In practice, companies first install transactional tools, because analytical tools need them to be fed with raw data.

More and more, multiagent systems are seen as a new technology for improving or replacing technologies used in both transactional and analytical information technologies. We now explain why agent technology seems so promising in the context of supply chains.

4.2 Using MultiAgent Systems in Supply Chain Management: Motivations

Some arguments in favor of using multiagent systems in supply chain management can be found in the literature. In fact, researchers have already applied agent technology in industry to concurrent engineering, collaborative engineering design, manufacturing enterprise integration, supply chain management, manufacturing planning, scheduling and control, material handling, and holonic manufacturing systems [47].

Concerning supply chains, Dodd and Kumara [79] think that Mark Fox (e.g. [80]) was probably the first to organize the supply chain as a network of intelligent agents. Indeed, supply chains are made up of heterogeneous production subsystems gathered in vast dynamic and virtual coalitions. Intelligent distributed systems, e.g. multiagent systems, enable increased autonomy of each member in the supply chain. Each partner (or production subsystem) pursues individual goals while satisfying both local and external constraints [81]. Therefore, one or several agents can be used to represent each partner in the supply chain (plant, workshop, etc.). Moreover, the agent paradigm is a natural metaphor for network organizations, since companies prefer maximizing their own profit than the profit of the supply chain [82]. In fact, the distributed manufacturing units have the same characteristics as agents [83] (based on Wooldridge [26]'s definition of agents, quoted previously):

- *autonomy*: a company carries out tasks by itself without external intervention and has some kind of control over its action and internal state;
- *social ability*: a company in the supply chain interacts with other companies, e.g. by placing orders for products or services;
- *reactivity*: a company perceives its environment, i.e., the market and the other companies, and responds in a timely fashion to changes that occur in it. In particular, each firm modifies its behaviour to adapt to market and competition evolutions;
- *pro-activeness*: a company not only simply acts in response to its environment, it can also initiate new activities, e.g. launching new products on the market;

Issue	Autonomous agents	Conventional systems
Model	Economics, biology	Military
Issues favouring conventional system		
1 Theoretical optima?	No	Yes
2 Level of prediction	Aggregate	Individual
3 Computational stability	Low	High
Issues favouring autonomous agents		
4 Match to reality	High	Low
5 Requires central data?	No	Yes
6 Response to change	Robust	Fragile
7 System reconfigurability	Easy	Hard
8 Nature of software	Short, simple	Lengthy, complex
9 Time required to schedule	Real time	Slow

Table 1. Agent-based vs. conventional technologies [84].

Moreover, multiagent systems offer a way to elaborate production systems that are decentralized rather than centralized, emergent rather than planned, and concurrent rather than sequential. Therefore, they allow relaxing the constraints of centralized, planned, sequential control [84]. Unfortunately, an agent-based approach is not a panacea for industrial softwares. Like other technologies, this approach has advantages and disadvantages: it must be used for problems whose characteristics require its capacities. According to Parunak [45], five characteristics are particularly salient. In fact, agents are best suited for applications that are *modular, decentralized, changeable, ill-structured* and *complex*.

To judge relevance for supply chains of autonomous agents, Parunak [84] compares this approch with conventional technologies in Table 1, thus highlighting differences between these two philosophies. To this end, multiagent systems are identified as biological (ecosystems) and economical (markets) models, whereas traditional approaches are compared with military patterns of hierarchical organization. Table 1 summarizes the main disadvantages of multiagent systems:

1. theoretical optima cannot be guaranteed, because there is no global view of the system;
2. predictions for autonomous agents can usually be made only at the aggregate level;
3. in principle, systems of autonomous agents can become computationally unstable, since, according to System Dynamics, any system is potentially unstable.

But on the other hand, the autonomous, agent-based approach has some advantages too:

4. because each agent is close to the point of contact with the real world, the systems's computational state tracks the state of the world very closely…

5. . . . and without need for a centralized database;
6. because overall system behaviour emerges from local decisions, the system readjusts itself automatically to environmental noise . . .
7. . . . or to the removal or addition of agents;
8. the software for each agent is much shorter and simpler than would be required for a centralized approach, and as a result is easier to write, debug and maintain.
9. because the system schedules itself as it runs, there is no separate scheduling phase of operation, and thus no need to wait for the scheduler to complete. Moreover, the optima computed by conventional systems may not be realizable in practice, and the more detailed predictions permitted by conventional approaches are often invalidated by the real world.

All these reasons show the relevance to use agents in supply chain management. In other words, thanks to their adaptability, their autonomy and their social ability, agent-based systems are a viable technology for the implementation of communication and decision-making in real-time. Each agent would represent a part of the decision-making process, hence creating a tight network of decision makers, who react in real-time to customer requirements, in opposition to the flood of current processes, which is decided before customers place an order [79].

4.3 Using MultiAgent Systems in Supply Chains: Examples

We now illustrate the use of agents in supply chains by presenting various projects. These projects can be separated into two broad families: supply chain management projects [85] and supply chain design projects. Moreover, the manner of solving problems also differs depending on projects, e.g. the number and the role of agents vary considerably, depending on the particular point under study. To highlight these differences, Table 2 summarizes the projects which are now described:

1. DragonChain was implemented by Kimbrough's team [86] at the University of Pennsylvania (Philadelphia, PA, USA) to simulate supply chain management, and more particularly to reduce the bullwhip effect. For that, they base their simulation on two versions of the *Beer Game*, the *MIT Beer Game* (i.e., the original game [102]) and the *Columbia Beer Game* [103], and they use agents that look for the best ordering scheme with genetic algorithms. Note that Chapter 10 uses simulations based on a model similar to the Beer Game, while Chapter 9 also applies genetic algorithms.
2. Agent Building Shell at the University of Toronto (Ontario, Canada) is a library of software classes providing reusable elements for building agent systems. These agents have four layers: a layer for knowledge management, an ontology layer, a layer of cooperation and conflict solving, and a layer of communcation and coordination. This latter layer is insured by the COOrdination Language (COOL). This project has involved several researchers, such as Mark Fox, Teigen, Barbuceanu and Beck [87–90].

Project	Studied problem	Approach	Number and role of agents
1. DragonChain [86]	Management (Bullwhip effect)	Genetic algorithm seeking the best ordering scheme	1 agent/company
2. Agent Building Shell [87–90]	Management (Coordination)	COOrdination Language (COOL)	1 agent/company
3. MetaMorph 1 & 2 [81]	Management (Coordination)	Mediator-agents	1 agent/company +mediator-agents
4. NetMan [83, 91]	Management (Intra- and inter-company operations management)	Contract driven coordination in Convention, Agreement and Transaction (CAT) formalism	1 agent/workshop
5. BPMAT &SCL [12, 92, 93]	Modelization (Which elements are common to all supply chains?)	Comparison of three very different supply chains	BPMAT models companies & SCL intercompany streams
6. MASCOT [94]	Management (Agility increase)	Comparison of several coordination policies	1 agent/company
7. DASCh [95, 96]	Management (supply chain modelization techniques)	Delays and uncertainties on streams modelled as agents	2 agents/company + 1 agent/stream
8.Task dependency network [97–99]	Design & management (Partner selection)	Comparison of auction protocols	1 agent/company
9. MASC [100]	Design (Partner selection)	Auction-based protocol under constraints	1 agent/company + 2 directory agents
10. OCEAN [101]	Management (Global cooperation emerging from local competitions)	Negotiation system in a multiagent contract network	1 agent/company (1 agent = system of 6 agents)

Table 2. Some projects applying agents to supply chains.

3. MetaMorph II is an improvement of a first project called MetaMorph. Agents form a federation centered around mediators that have two roles: they allow agents to find each other, and they coordinate these agents. These two projects were developed at the University of Calgary (Alberta, Canada) by Maturana and others [81].
4. NetMan (NETworked MANufacturing) formalizes networked organizations and production operations in order to obtain agile manufacturing networks in a dynamic environment. Conversely to DragonChain, this multiagent system manages an actual supply chain, rather than the Beer Game. Each company is cut in NetMan centers, i.e., independent, collaborating business units. The NetMan centers of a company coordinate with each other and with other customers' and suppliers' NetMan centers. This coordination is based on contracts and conventions, which are formalized according to the model *Convention, Agreement, Transaction* (CAT). This work was carried out at Université Laval (Quebec City, Quebec, Canada) [83, 91]. Chapters 14, 15 and 16 detail additional industrial applications of agents.
5. BPMAT is a software library developed by IBM [93] to model company activities (note that Chapter 15 presents another project of IBM). SCL is an

addition to this library for modelling inter-company flows. BPMAT and SCL are based on Swaminathan, Smith and Sadeh [12, 92]'s work at Carnegie Mellon University (Pittsburgh, PA, USA), which sought elements common to any supply chain by comparing three chains from distinct industrial sectors. Chapter 8 also deals with supply chain modeling.

6. MASCOT (MultiAgent Supply Chain cOordination Tool) is a reconfigurable, multilevel, agent-based architecture for planning and scheduling aimed at improving supply chain agility. It coordinates production among multiple (internal or external) facilities, and evaluates new product/subcomponent designs and strategic business decisions (e.g., make-or-buy or supplier selection decisions) with regard to capacity and material requirements across the supply chain [94]. Like BPMAT and SCL, this work was also accomplished at Carnegie Mellon University (Pittsburgh, PA, USA).
7. DASCh was developed at ERIM (Ann Arbor, MI, USA) by Parunak and his colleagues [95, 96] to explore the modelling techniques of networks of suppliers and suppliers' suppliers. In particular, flows of products and information flows are viewed as agents to model imperfections in these flows. The approach of risk management in Chapter 4 may manage such imperfections as well.
8. The Task Dependency Network is an asynchronous, decentralized market protocol (auctions) for allocating and scheduling tasks among agents that contend for scarce resources, constrained by a hierarchical task dependency network [97, 99]. An additional paper [98] extends this protocol to model supply chain formation. This work is a Ph.D. thesis defended in 2001 by Walsh [97] (supervised by Wellman) at the University of Michigan (Ann Arbor, MI, USA). Walsh presents a more recent work with Babaioff in Chapter 12.

   In similar ways, other works use market mechanisms to coordinate supply chains [104, 105].
9. MASC studies coordination modes between companies in supply chains. These coordination modes are calls for submissions, which submitters answer according to their capacity and production load. Companies winning this auction next take part in the supply chain carrying products to the consumer. This work was completed at the Université d'Aix-Marseille 3 (Marseilles, France) [100]. Chapters 2, 3, 5, 11 and 13 (i.e., those about e-commerce and negotiation) also address these questions about where and how many items to buy.
10. OCEAN (Organization and Control Emergence with an Agent Network) is a control system with an open, decentralized and constraints-based architecture in which there is responsiveness, and distribution of production resources and technical data. This system was designed to react to environment dynamics in order to show that cooperation at the global level may emerge from competitions at the local level. This work was completed at INSA de Lyon (Lyon, France) and at the Université de Montpellier 2 (Montpellier, France) [101]. Chapters 7 and 8 also treat environment dynamics.

5 Conclusion

The literature review in this chapter has introduced the topic of this book by presenting the two areas involved and their merger. Specifically, the first area involved, i.e., supply chain management, was first introduced by outlining some industrial problems, and how the concept of supply chain as well as the collaboration it enables allow solving these problems.

After that, we turned to multiagent systems as a way to implement the supporting technologies required by the concept of supply chain. We described what an autonomous agent is, and how agents are different from objects. We also presented some agent architectures, the motivation for the use of agents in any field, and a comparison of multiagent systems with other areas. Some applications of multiagent systems illustrated what multiagent systems are, followed by a focus on one specific application, that is, on multiagent-based supply chain management.

This focus on the use of agents in supply chains first detailed the information technologies required by supply chains, next focussed the previous motivation for agents on their application to supply chains. It is important to note that what supply chains require is the main characteristic of agents, i.e., their autonomy. Finally, the summary of some projects using agents in supply chains illustrated this chapter.

Similarly to this illustration, the rest of this book contains additional applications of agents to supply chain management or to closely related areas, such as e-commerce and risk management in supply chains.

6 Acknowledgment

We would like to thank FOR@C, the Research Consortium in E-business in the forest products industry (Université Laval, Québec, Canada), for supporting this research. This work was also partially supported by the National Sciences and Engineering Research Council of Canada (NSERC).

References

1. Mallya, S.N.: Demand Diven Production and Distribution: Seamless Integration of Downstream Supply Chain Decisions using Electronic Data Interchange. Ph.D. thesis, Drexel University (Philadelphia, PA, USA) (1999)
2. Savelsbergh, M.W.P.: Local search in routing problems with time windows. Annuals of Operations Research **4** (1985) 265–305
3. Simchi-Levi, D., Kaminsky, P., Simchi-Levi, E.: Designing and managing the supply chain. McGraw-Hill Higher Education (2000)
4. Shapiro, J.F.: Modeling the Supply Chain. Duxbury (2000)
5. Lee, H.L., Padmanabhan, V., Whang, S.: The bullwhip effect in supply chain. Sloan Management Review **38** (1997) 93–102
6. Lee, H.L., Padmanabhan, V., Whang, S.: Information distortion in a supply chain: The bullwhip effect. Management Science **43** (1997) 546–558

7. Carlsson, C., Fullér, R.: Reducing the bullwhip effect by means of intelligent, soft computing methods. In: Proc. 34th Hawaii International Conference on System Sciences (HICSS-34), (Hawaii, USA) (2001)
8. Iaccoca Institute: 21st Century Enterprise Strategy: An Industry Led View (1991) Lehig University, Bethlehem (Pennsylvania, USA).
9. Rigby, C., Day, M., Forrester, P., Burnett, J.: Agile supply: Rethinking systems thinking, systems practice. International Journal of Agile Management Systems **2** (2000) 178–186
10. Burns, J.F., Sivazlian, B.D.: Dynamic analysis of multi-echelon supply systems. Computers & Industrial Engineering **2** (1978)
11. Muckstadt, J., Murray, D., Rappold, J., Collins, D.: Guidelines for collaborative supply chain system design and operation. Information Systems Frontiers **3** (2001) 427–435
12. Swaminathan, J.M., Smith, S.F., Sadeh, N.M.: Modeling supply chain dynamics: A multiagent approach. Decision Sciences **29** (1998) 607–632
13. Davidsson, P., Werstedt, F.: A multi-agent system architecture for coordination of just-in-time production and distribution. Knowledge Engineering Review **17** (2002) 317–329
14. Poirier, C.C., Reiter, S.E.: La Supply Chain - Optimiser la Chaîne Logistique et le Réseau Interentreprises. Dunod (Les Réalités de l'entreprise) (2001) Translation of "The Supply Chain Optimization: Building the Strongest Total Business Network" (1996) by Monique Sperry.
15. Chen, F., Drezner, Z., Ryan, J.K., Simchi-Levi, D.: The bullwhip effect: Managerial insights on the impact of forecasting and information on variability in a supply chain. In: Quantitative Models for Supply Chain Management. Sridhar Tayur, Ram Ganeshan and Michael Magazine (1994)
16. Taras, J.: VMI Web site (2003) `http://www.vendormanagedinventory.com/` (accessed Sept. 8, 2003).
17. Wal-Mart Stores, Inc.: Web site (2004) `http://www.walmart.com/` (accessed April 5, 2004).
18. Procter & Gamble Company: Web site (2004) `http://www.pg.com/` (accessed April 5, 2004).
19. Waller, M., Johnson, M.E., Davis, T.: Vendor-managed inventory in the retail supply-chain. Journal of business logistics **20** (1999) 183–203
20. VICS Association: Web site (2003) `http://www.cpfr.org` (accessed Sept. 10, 2003).
21. Lee, H.L.: E-business and supply chain integration. Stanford Global Supply Chain Forum `http://www.stanford.edu/group/scforum/Welcome/EB_SCI.pdf` (2001)
22. Chaib-draa, B., Moulin, B., Jarras, I.: Agent et systèmes multiagents. In J.P. Briot et Y Demazeau, ed.: Principes et Architecture des Systèmes Multi-Agents. Hermes, Lavoisier (2001)
23. Chaib-draa, B., Moulin, B., Mandiau, R., Millot, P.: Trends in distributed artificial intelligence. Artificial Intelligence Review **6** (1992) 35–66
24. Chaib-draa, B.: Industrial applications of Distributed AI. Communication of ACM **38** (1995) 49–53
25. Ferber, J.: Les systèmes multi-agents, vers une intelligence collective. InterEditions, Paris (France) (1995)
26. Wooldridge, M., Jennings, N.R.: Intelligent agents: Theory and practice. The Knowledge Engineering Review **10** (1995) 115–152
27. Shoham, Y.: Agent-oriented programming. Artificial Intelligence **60** (1993) 51–92

28. Free Software Foundation: Web site of GCC (GNU compiler collection) (2004) http://gcc.gnu.org/ (accessed April 14, 2004).
29. Sun microsystems: Java 2 platform (2003) http://java.sun.com/ (accessed Sept. 19, 2003).
30. Wooldridge, M.: An Introduction to Multi-Agent Systems. John Wiley & Sons, Inc. New York, NY, USA (2001)
31. Wooldridge, M.: Intelligent agents. In Weiss, G., ed.: Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence. MIT Press, Cambridge (Massachusetts, USA) (1999) 28–77
32. Guessoum, Z., Briot, J.P.: From active objects to autonomous agents. IEEE Concurrency **7** (1999)
33. Gasser, L., Briot, J.P.: Object-based concurrent programming and distributed artificial intelligence. In Avouris, N., Gasser, L., eds.: Distributed Artificial Intelligence: Theory and Praxis. Kluwer Academic Publishers (1992)
34. Agent Oriented Software Group: JACK Intelligent Agent (TM) User Guide. Technical Report Release 4.1 (2003) http://www.agent-software.com/.
35. Russell, S., Norvig, P.: Artificial Intelligence - A modern approch. Englewood Cliffs, NJ: Prentice-Hall (2d Edition) (2003)
36. Turing, A.M.: Computing machinery and intelligence. Mind **59** (1950) 433–460
37. Huhns, M.N., Stephens, L.M.: Multi-agent systems and societies of agents. In Weiss, G., ed.: Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence. MIT Press, Cambridge (Massachusetts, USA) (1999) 79–120
38. Jennings, N.: On agent-based software engineering. Artificial Intelligence **117** (2000)
39. Jennings, N., Sycara, K., Wooldridge, M.: A roadmap of agent research and development. International Journal of Autonomous Agents and Multi-Agent Systems **1** (1998) 7–38
40. Papadimitriou, C.H.: Algorithms, games, and the internet. In: Proc. STOC 2001 (extended abstract in ICALP 2001). (2001)
41. Boutilier, C., Shoham, Y., Wellman, M.P.: Economic principles of multi-agent systems (editorial). Artifical Intelligence **94** (1994)
42. Jennings, N., Wooldridge, M.: Applications of intelligent agents. In: Agent technology: Foundations, applications and markets. Springer-Verlag New York, Inc. (1998) 3–28
43. Weiss, G.: Multiagent systems - A modern approach to distributed artificial intelligence. The MIT Press (1999)
44. Gruver, W.A., Kotak, D.B., van Leeuwen, E.H., Norrie, D.H.: Holonic manufacturing systems: Phase 2. In: Holonic and Multi-Agent Systems for Manufacturing - 1st International Conference on Industrial Applications of Holonic and Multi-Agent Systems, HoloMAS 2003, Prague (Czech Republic) http://gerstner.felk.cvut.cz/HoloMAS/2003/ (2003) 1–14
45. Parunak, H.V.D.: Industrial and practical applications of DAI. In Weiss, G., ed.: Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence. MIT Press, Cambridge (Massachusetts, USA) (1999) 377–424
46. Shen, W., Norrie, D.H.: Agent-based systems for intelligent manufacturing: A state-of-the-art survey. Knowledge and Information Systems **1** (1999) 129–156
47. Shen, W., Norrie, D.H., Barthès, J.P.A.: Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing. Taylor & Francis (London and New York) (2001)
48. British Telecommunications plc: The ZEUS agent building toolkit (2004) http://more.btexact.com/projects/agents/zeus/ (accessed May 21, 2004).

49. Ljungberg, M., Lucas, A.: The OASIS air traffic management system. In: Proc. 2nd Pacific Rim International Conference on Artificial Intelligence (PRICAI '92). (1992)
50. Callantine, T.J.: Air traffic controller agents. In: Proc. 2nd int. joint conf. on Autonomous Agents and MultiAgent Systems (AAMAS), Melbourne (Victoria, Australia) (2003) 952–953
51. Cammarata, S., McArthur, D., Steeb, R.: Strategies of cooperation in distributed problem solving. In: Proc. 8th International Joint Conference on Artificial Intelligence (IJCAI-83). (1983) 767–770
52. Hallé, S., Chaib-draa, B., Laumonier, J.: Car platoons simulated as a multiagent system. In: Proc. 4th Workshop on Agent-Based Simulation, Montpellier (France), SCS (2003) 57–63
53. eBay, Inc.: Web site (2004) `http://www.ebay.com/` (accessed May 16, 2004).
54. Ben-Ameur, H., Chaib-draa, B., Kropf, P.: Multi-items auctions for automatic negotiation. Journal of Information and Software Technology **44** (2002) 291–301
55. Rosenschein, J.S., Zlotkin, G.: Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers. MIT Press, Cambridge (Massachusetts, USA) (1994)
56. Asselin, F.: Formation de coalitions d'agents logiciels pour le groupement d'acheteurs. Master's thesis, Université Laval, Département d'Informatique et de Génie Logiciel, Quebec City (Quebec, Canada) (2002)
57. Asselin, F., Chaib-draa, B.: Towards a protocol for the formation of coalitions of buyers. In Crainic, T.G., Gavish, B., eds.: Proc. Fifth International Conference on Electronic Commerce Research (ICECR-5). (2002)
58. Breban, S., Vassileva, J.: A coalition formation mechanism based on inter-agent trust relationships. In: Proc. 1st int. joint conf. on Autonomous Agents and MultiAgent Systems (AAMAS), Bologna, Italy (2002) 306–307
59. Trading Agent Competition: Web site (2004) `http://www.sics.se/tac/` (accessed May 16, 2004).
60. Cheng, S.F., Leung, E., Lochner, K.M., O'Maley, K., Reeves, D.M., Schvartzman, L.J., Wellman, M.P.: Walverine: A walrasian trading agent. In: Proc. 2nd int. joint conf. on Autonomous Agents and MultiAgent Systems (AAMAS), Melbourne (Victoria, Australia) (2003) 465–472
61. Vetsikas, I.A., Selman, B.: A principled study of the design tradeoffs for autonomous trading agents. In: Proc. 2nd int. joint conf. on Autonomous Agents and MultiAgent Systems (AAMAS), Melbourne (Victoria, Australia) (2003) 473–480
62. Grand, S., Cliff, D.: Creatures: Entertainment software agents with artificial life. Autonomous Agents and Multi-Agent Systems **1** (1998)
63. Hayes-Roth, B., Hewitt, M., Washington, R., Seiver, A.: Distributing intelligence with an individual. In Gasser, L., Huhns, M., eds.: Distributed Artificial Intelligence, Vol. II. Kaufmann (San Mateo, California, USA) (1989) 385–412
64. Hiroaki, K.: Robocup rescue: A grand challenge for multi-agent systems. In: Proc. International Conference on Multi-Agent Systems (ICMAS 2000). (2000)
65. Paquet, S.: Learning Coordination in RoboCupRescue. In Xiang, Y., Chaib-draa, B., eds.: Lecture Notes in Artificial Intelligence. Volume 2671., Springer-Verlag (2003) 627–628 Proc. 16th Conference of the Canadian Society for Computational Studies of Intelligence (AI 2003).
66. Microsoft Corp.: Pack MS-Office 2000 (2004) `http://www.microsoft.com/office/` (accessed Jan. 29, 2004).

67. Franchesquin, N., Espinasse, B.: Analyse multiagent de la gestion hydraulique de la camargue: Considérations méthodologieques. In: Proc. Journées Francophones des SMA et de l'IAD, St Etienne (France) (2000) 325–342
68. Cingil, I., Dogac, A.: An architecture for supply chain integration and automation on the internet. Journal of Distributed and Parallel Databases **10** (2001) 59–102
69. World Wide Web Consortium: Web site (2003) http://www.w3c.org (accessed Sept. 10, 2003).
70. World Wide Web Consortium: Specification of the Resource Description Framework (RDF) 1.0 (2003) www.w3.org/TR/rdf-schema/ & www.w3.org/TR/REC-rdf-syntax (accessed Sept. 10, 2003).
71. World Wide Web Consortium: Specification of the web ontology language (OWL) (2004) http://www.w3.org/2004/OWL/ (accessed March 31, 2005).
72. OASIS: The XML cover pages - XML Common Business Library (xCBL) (2003) www.oasis-open.org/cover/cbl.html (accessed Sept. 19, 2003).
73. Singh, M., Huhns, M.N.: Service-Oriented Computing - Semantics, Processes, Agents. John Wiley and sons, Ltd. (2005)
74. Garé, B.: Entreprise, ouvre-toi ! Logiciels & systèmes **52** (2000) Dossier spécial chaîne logistique.
75. Oracle: Web site (2005) http://www.oracle.com/peoplesoft/integration.html (accessed Oct. 5, 2005).
76. SSA Global: Web site (2005) http://www.ssaglobal.com/ (accessed Oct. 5, 2005).
77. Commerce One, LLC: Web site (2005) http://www.commerceone.com/ (accessed Oct. 7, 2005).
78. Ariba, Inc.: Web site (2003) http://www.ariba.com (accessed Sept. 5, 2003).
79. Dodd, C., Kumara, S.R.T.: A distributed multi-agent model for value nets. IEA/AIE 2001 (2001) 718–727
80. Fox, M.S., Chionglo, J.F., Barbuceanu, M.: The integrated supply chain management (1993) Internal report of the Enterprise Integration Laboratory, Department of Industrial Engineering, University of Toronto (Ontario, Canada).
81. Maturana, F., Shen, W., Norrie, D.: Metamorph: an adaptive agent-based architecture for intelligent manufacturing. International Journal of Production Research **37** (1999) 2159–2173
82. Viswanathan, S., Piplani, R.: Coordinating supply chain inventories through common replenishment epochs. European Journal of Operational Research **129** (2001) 277–286
83. Cloutier, L., Frayret, J.M., D'Amours, S., Espinasse, B., Montreuil, B.: A commitment-oriented framework for networked manufacturing coordination. International Journal of Computer Integrated Manufacturing **14** (2001) 522–534
84. Parunak, H.V.D.: Applications of distributed artificial intelligence in industry. In O'Hare, G.M.P., Jennings, N.R., eds.: Foundations of Distributed Artificial Intelligence. John Wiley & Sons (1996) 71–76
85. Chen, Y., Peng, Y., Finin, T., Labrou, Y., Cost, S.R.: Negotiating agent for supply chain management (1999)
86. Kimbrough, S.O., Wu, D., Zhong, F.: Computers play the Beer Game: Can artificial agents manage supply chains? Decision Support Systems **33** (2002) 323–333
87. Teigen, R., Barbuceanu, M.: The supply chain demonstrator (1996) Technical report and user guide by the Enterprise Integration Laboratory, Department of Industrial Engineering, University of Toronto (Ontario, Canada).

88. Barbuceanu, M., Fox, M.: The architecture of an agent building shell. In: Intelligent Agents II, Springer-Verlag, LNAI 1037. (1995) 235–250
89. Barbuceanu, M., Fox, M.: Cool: A language for describing coordination in multiagent systems. In: Proc. First International Conference on Multi-Agent Systems. (1995)
90. Beck, J., Fox, M.: Supply chain coordination via mediated constraint relaxation. In: Proc. 1st Canadian Workshop on Distributed Artificial Intelligence, Banff (Alberta, Canada) (1994) 61–72
91. Lyonnais, P., Montreuil, B.: Un environnement distribué de simulation à base d'agents. In: 3e conférence francophone de MOdélisation et SIMulation (MOSIM) - conception, analyse et gestion des systèmes industriels, Troyes (France) (2001)
92. Swaminathan, J.M., Smith, S.F., Sadeh, N.M.: Modeling the dynamics of supply chains. In: Proc. AAAI-94 Workshop on reasoning about the shop floor, Seattle (Washington, USA) (1994)
93. IBM Research: BPMAT: A business process modeling and analysis tool (2003) `http://www.research.ibm.com/pdtr/bpmat.htm` (accessed Sept. 19, 2003).
94. Sadeh, N., Hildum, D., Kjenstad, D., Tseng, A.: MASCOT: An agent-based architecture for coordinated mixed-initiative supply chain planning and scheduling. In: Workshop notes, Agent-Based Decision Support for Managing the Internet-Enabled Supply Chain, 3rd International Conference on Autonomous Agents (Agents' 99). (1999)
95. Parunak, H.V.D., VanderBok, R.: Modeling the extended supply network. In: ISA-Tech'98 (Houston, Texas, USA), Industrial Technology Institute (1998)
96. Baumgaertel, H., Brueckner, S., Parunak, V., Vanderbok, R., Wilke, J.: Agent models of supply network dynamics. The Practice of Supply Chain Management (2001) URL http://www.anteaters.net/ sbrueckner/publications/2001/-AgentModelsOfSupplyNetworkDynamics_submitted_20010209.pdf.
97. Walsh, W.E.: Market protocols for decentralized supply chain formation. Ph.D. thesis, University of Michigan, Ann Arbor (Michigan, USA) (2001)
98. Walsh, W.E., Wellman, M.P.: Modeling supply chain formation in multiagent systems. In: Agent mediated electronic commerce (IJCAI workshop). (1999) 94–101
99. Walsh, W.E., Wellman, M.P.: A market protocol for decentralized task allocation. In: Proc. 3rd International Conference on Multi-Agent Systems (ICMAS-98), Department of computer science and information systems, University of Hong Kong (1998)
100. Labarthe, O.: Modélisation multi-agent de chaînes logistiques. Mémoire de Diplôme d'Études Approfondies, Université d'Aix-Marseille 3, Marseilles (France) (2000)
101. Bournez, C., Gutknecht, O.: Modèle de contrôle par émergence de coordinations dans un réseau de contrats multiagents. In: Journées Francophones de l'Intelligence Artificielle Distribuée et des Systèmes MultiAgents, Montreal (Quebec, Canada) (2001)
102. Sterman, J.D.: Modeling managerial behavior: Misperceptions of feedback in a dynamic decision making experiment. Management Science **35** (1989) 321–339
103. Chen, F.: Decentralized supply chains subject to information delays. Management Science **45** (1999) 1076–1090
104. Eymann, T.: Markets without makers - A framework for decentralized economic coordination in multiagent systems. In: Lecture Notes in Computer Science 2232. (2001) 63–74 Proc. 2nd International Workshop WELCOM.

105. Fan, M., Stallaert, J., Whinston, A.B.: Decentralized mechanism design for supply chain organizations using an auction market. Information Systems Research **14** (2003) 1–22