Multi-Behavior Agent Model for Planning in Supply Chains: An Application to the Lumber Industry

Pascal Forget

Sophie D'Amours

Jean-Marc Frayret

November 2006

Working Paper DT-2006-SD-03

Interuniversity Research Center on Enterprise Networks, Logistics and Transportation (CIRRELT)

Université Laval, Québec, G1K 7P4 CANADA

FOR@C

DE LA FORÊT AU CLIENT

CIRRELT

Abstract — Recent economic and international threats to western industries have encouraged companies to increase their performance in any manner possible. Many look to deal quickly with disturbances, reduce inventory and exchange information promptly throughout the supply chain. In other words they want to become more agile. To reach this objective it is critical for planning systems to present planning strategies adapted to the different contexts, to attain better performances. The development of integrated supply chains and the use of inter-organizational information systems have increased business interdependencies and in turn the need for increased collaboration to deal with disturbance in a synchronized way. Thus, agility and synchronization in supply chains are critical to maintain overall performance. In order to develop tools to increase the agility of the supply chain and to promote the collaborative management of such disturbances, agent-based technology takes advantage of the ability of agents to make autonomous decisions in a distributed network through the use of advanced collaboration mechanisms. Moreover, because of the highly instable and dynamic environment of today's supply chains, planning agents must handle multiple problem solving approaches. This paper proposes a Multi-behavior planning agent model using different planning strategies when decisions are supported by a distributed planning system. The implementation of this solution is realized through the FOR@C experimental agent-based platform, dedicated to supply chain planning for the lumber industry.

Keywords — Supply chain management, agent model, agent-based planning systems, advanced planning system (APS), lumber industry.

1. Introduction

Recent economic and international threats to western industries have encouraged companies to increase their performance in any possible way. Many look to rethink their planning systems in a way to quickly react to and correct deviance from established plans, respond to demand, reduce inventory and exchange information promptly throughout the supply chain [20]. In other words companies want to become more agile. Agility can be described as the association of flexibility, which is the ability to react to changes by presenting different solutions, and high responsiveness, which is the ability to react in a timely fashion. To reach this objective it is critical for planning systems to present planning strategies adapted to the different contexts in order to reach better overall performances. Due to consolidation, the development of integrated supply chains and the use of inter-organizational information systems have increased business interdependencies and in turn the need for increased collaboration to deal with disturbance in a synchronized way. Global organization forces have recognized that performance is not the feature of a single firm, but the complex output of a network of interconnected firms [28]. Thus, agility and synchronization in supply chains are critical to maintain overall performance. Efforts have been deployed to increase supply chain performance as a way to stay competitive with international consortiums. Developed mainly to improve efficiency between partners by increasing coordination and communication, supply chain management has been studied in multiple ways [e. g. 43, 45].

For years supply chains have been (and are still mostly) managed in a hierarchical way, where demand plans (customer orders in a context of dynamic demand) are calculated locally and transmitted to suppliers. This sequential planning gives full autonomy to each company and organizational unit involved, but no effort is invested in synchronizing plans and using partner capacity. In fact, the only synchronization tool is the actual demand plan sent to suppliers in order to improve demand forecast and reduce the bullwhip effect [see 25].

The distributed decision making paradigm provides an interesting approach to increase agility by permitting local correction of the plan, while promoting a global coherence in the supply chain. This is done by keeping planning decisions distributed, yet using close collaboration mechanisms between organizational units to ensure synchronization of production plans. Agent-based technology provides a natural platform that takes advantage of the autonomy of agents and their ability to make decisions in a distributed context, using collaboration and goal-driven decisions. A distributed agent-based

Advanced Planning and Scheduling system (d-APS) could maintain a real-time plan by re-planning locally and allow for collaboration between agents to deal with disturbances.

At the same time, the highly instable and dynamic environments of supply chains ask for an increased ability for planning systems to correct deviance from disturbance in an adapted way. This can be possible by increasing the intelligence of planning agents, in order to give them sufficient competencies to use the right strategy for the right situation. There is a need to clarify what competencies are needed and how they can be used in an agent-based system to show efficient behaviors to react promptly to disturbances and to correct unwanted situations.

In this paper, Section 2 provides a literature review on supply chain planning and how disturbances are handled in such complex environments. Different uses of agent-based technologies in supply chains and different agent architectures proposed in literature are presented. Then, Section 3 describes the Collaborative Event Management approach, which proposed how collaboration between production units could be used to deal efficiently with disturbances. In Section 4, we explain the experimental agent-based planning platform developed by the FOR@C Research Consortium, which is dedicated to supply chain planning for the lumber industry. Our contributions to the literature are presented in Sections 5, 6 and 7. In Section 5, an agent conceptual model is presented, geared with tools designed to improve agility and synchronization in supply chains. Section 6 details the Multi-behavior agent model, which is an extension of our conceptual model. Section 7 describes a possible implementation of the agent, using different planning protocols, in order to give an idea of the full potential of the agent. We describe briefly how we plan to simulate and test the agent model. Finally, in Section 8, we present our conclusion.

The North American lumber industry represents a perfect context for this agent model. In fact, this industry is highly distributed, with many production units interacting in all activity levels. This industry is characterized by stochastic disturbances in many aspects of its supply chain, mainly due to the heterogeneous aspect of the resource, uncertain process output, production of co-products and by-products, price variation in the spot market and demand variation in commodity markets.

2. Literature review

2.1. Planning in supply chains

Supply chain planning involving different companies represent an important challenge. Partners do not exchange private information easily and are reluctant to share a common database [43]. When organizational units are part of the same company, which can be called an internal supply chain or intra-organizational supply chain, centralized information and planning systems are sometimes used. Gathering information in a centralized management system and redistributing plans can ensure synchronization and optimization of plans. Decision support systems, such as Advanced Planning and Scheduling (APS) systems are sophisticated sets of decision support applications using operational research (OR) techniques to find optimal solutions to complex planning problems [18]. However, even in an internal supply chain, when the number of organizational units grows, planning problems become more complex and hard to handle in a centralized way. Also, because of the quantity of information only available locally and the time it takes to plan the entire supply chain, plans are sometimes not feasible and the supply chain demonstrates low reactivity. In fact, currently available software solutions generally do not provide the necessary support to network organizations and are clearly insufficient in planning and coordinating activities in heterogeneous environments [6, 43]. Moreover, planning, scheduling and traditional control mechanisms are insufficiently flexible to react to rapid changes in production modes and client needs [27]. In other words, traditional systems have not been developed to work in decentralized, dynamic and heterogeneous environments.

In recent years there has been a trend of new management systems emerging. Because coordination cannot be implicitly transmitted from a top level, collaboration and coordination mechanisms are needed to insure synchronization and consistency throughout the supply chain. This opened the way to an entire new research domain, which is supply chain management (SCM), where researchers are interested in coordination and decision making between supply chain partners to optimize the supply chain performance [45].

2.2. Dealing with disturbances in supply chains

A major difficulty in supply chain planning is dealing with disturbances in an efficient way. In fact, disruptions and uncertainties have been a problem since the beginning of systemized manufacturing and remains an important subject [5]. Disturbances can take different forms, such as change in demand, machine breakdown, late delivery, employee sickness, etc. In a dynamic environment, as in a production plant, as soon as a plan is released, it is immediately subject to random disruptions that quickly render the initial plan obsolete [2]. The traditional way to avoid disturbance related problems is to keep inventories. In fact, inventory exists as an insurance against uncertainty [13]. While costly, this approach considerably reduces flexibility, because stocked products must be sold even if demand has changed. In contrast, less stock means reducing the overall inventory investment, freeing up available cash flow and improving end-customer service [13]. Keeping low inventory requires close collaboration with partners to ensure precise information on needs.

Companies experience business interdependencies in their day to day operations. They therefore acknowledge that the behavior of one can influence another. In a highly dependent network of entities, when activities are tightly planned, disturbances can have important repercussions throughout the supply chain. For example, a major mechanical breakdown in a strategic third-tier supplier can reduce supply availability for several days, which can have tremendous impacts on the whole supply chain, translating in a delay for the final client. Another example is a quick change in demand pattern. When such change happens, every demand plan exchanged between each partner must be updated. If it is not done in a very short period of time, inventories will pile-up and money will be wasted. To counter these problems and their repercussions on the supply chain, CPFR (Collaborative Planning, Forecasting & Replenishment) methodologies are used and forecasts are prepared jointly.

Much work has been done for dealing with disturbances and uncertainty in a production context. Aytug et al. [5] present a literature review on production scheduling facing uncertainties in the context of a shop floor. Some researchers have presented works on Reactive Scheduling [e.g. 24], which is dedicated to the continuous adaptation of the schedule in a real-time context, with the objective of minimizing perturbations to the initial schedule. Confronted with disturbances, other researchers have worked on finding approaches to modify plans while minimizing impacts on performance using OR techniques [e.g. 2, 3, 7] and artificial intelligence (AI) techniques [e.g. 39]. Replanning is about repairing or starting a new plan in order to adapt to a new context. Robust scheduling is another approach to deal with disturbances, where the objective is to build a schedule with the best worst-case performance [e.g. 12]. Authors have also presented classifications, management frameworks and planning system requirements to deal efficiently with disturbances [11, 13, 17, 33].

2.3. Agent-based system in supply chains

A new trend of distributing decisions has resulted in the development of planning systems with agent-based architectures. These approaches are rooted in multi-agent technologies, coming from the AI domain [46]. Agent-based systems focus on implementing individual and social behaviors in a distributed context, using notions like autonomy, reactivity and goal-directed reasoning [9]. The emergence of agent-based systems has represented a real breakthrough in the research world, including researchers from various domains, such as biology, sociology, transportation, management, production, logistics and the military. Agent-based systems are computer systems made from a collection of agents, defined as intelligent software with specific roles and goals, interacting with each other to make the most appropriate decision according to the situation, in order to carry out their part of the planning task [26]. Distributed planning demonstrates many advantages over central planning. For complex problems, subproblems are easier to solve than centralized problems. Because decisions are distributed to different entities, reactivity to changes is increased. Also, due to the fact that local problems are smaller, it is possible to add more details in resolution, which is likely to improve feasibility of plans. The challenge here is that global supply chain performance is linked to agent collaboration capabilities to find acceptable compromises, insuring synchronization of plans.

Agent-based technology has already been applied to different areas in supply chain management. Parunak [31] provides a taxonomy of agent technology applied to industrial use and how it can be used in an operational context. He presents a dozen of case studies involving industrial agent-based systems, describing types of agents used, structure, protocols and maturity. Shen & Norrie [41] describe more than 30 research projects addressing scheduling, planning and control. More recently, Caridi et al. [10] present a survey and a classification of the different application domains of published multi-agent projects, denoting their degree of maturity. More specifically, agent-based planning systems have been proposed to manage supply chains and deal with disturbances. Montreuil et al. [28] present the NetMan architecture, an operation system for networked manufacturing organizations that aims to provide a collaborative approach to operations planning. Although the authors created an architecture able to manage unplanned events, they do not present specific behaviors to solve problems following disturbances. Based on intelligent *holons*, Fletcher et al. [16] present a conceptual architecture of a lumber processing system to improve flexibility and fault tolerance. The ProPlanT multi-agent platform [32] gives decision-making support and simulation possibilities to distributed production planning. With communication agents, project planning agents, project management agents and production agents, they use negotiation, job delegation and task decomposition instead of classic planning and scheduling mechanisms. In order to reduce communication traffic, social knowledge is precompiled and maintained, which represents information about other agents. Building on these research works, we propose to extend the representation of coordination mechanisms in order to increase supply chain agility and synchronization.

2.4. Agent architectures

Agents can be designed in various ways. The architecture of an agent has a direct impact on his behavior and how he reacts when confronted with different situations. Several classifications of architectures are proposed in the literature [e.g. 9, 42]. Basically, three main architectures are prominent: reactive, deliberative and hybrid agents. Reactive and deliberative agents represent extreme cases of behaviors, whereas hybrid agents are positioned somewhere between the two.

A reactive architecture basically links specific inputs to specific outputs. For example, for a specific observation in the environment, the agent has a pre-determined action. These agents have no internal representation of their world and no symbolic representation of knowledge. Although this architecture can perform very well in simple environments, an agent can show a lack of intelligence and adaptability in a more complex world. An evolved reactive architecture is presented by Brooks [8], which is the subsumption architecture, also called behavior-based architecture. Instead of a single specific reaction to an input, the reactive agent is decomposed into behaviors which are small independent processes that can be triggered, and where some cancel others. Instead of implementing a simple reactivity mechanism, the agent shows an emergent intelligent behavior, resulting from adaptation to his environment. The main advantage of this architecture is the fast adapted response, because no complex processing is needed. The disadvantage is the difficulty in creating oriented behaviors that follow long term goals and strategies.

In contrast, deliberative agents use their knowledge about their environment and their internal goals to plan and execute actions. They translate information from the world into symbolic knowledge, which they use to update their internal data base. The BDI (Belief-Desire-Intention) architecture [34] is a well-known example of a deliberative architecture, where the agent uses his knowledge about the world (belief) and his goals (desire) to build a plan of action (intention). The advantage of this architecture is the possibility to plan a sequence of actions, in order to meet long term goals. The agent can understand a complex environment and take an appropriate decision following a set of specific inputs. The disadvantage is the slow reaction time in dynamic environments, where situations can change while the agent is processing to find a suitable action. Also, the problematic of knowledge representation is complex and represents an entire research domain where researchers have been studying new approaches for years [e.g. 30].

Hybrid agents fit in between these extremes to find an optimal balance of these behaviors. Many authors have presented such architectures. The InteRRaP architecture [29] is a layered-based model, composed of three different layers: a behavior-based layer, a local planning layer and a cooperative layer. For a new situation, the agent first tries to find a rule in the behavior-based layer, which represents the reactive part of the agent. If no rule is known, the agent uses his second layer, the local planning layer, where a plan is built to solve the problem, following his own goals. If no solution is found, the agent uses his last layer, the cooperative layer, where he collaborates with other agents to find a feasible solution. Hybrid agents try to compile advantages of both reactive and deliberative architectures, using the best behavior in each situation. The main challenge of this architecture resides in the difficulty for the designer to co-ordinate the different layers in order to see an emergent coherent and intelligent agent behavior [9].

2.5. Hybrid agent architecture in supply chain planning

Several architectures and agent models have been adapted in supply chain context, specifically to improve supply chain performance by planning activities across business units and reacting to disturbances. The variety of possible disturbances, their stochastic distribution and their interactions make supply chain planning a highly complex task, raising the need for deliberative behavior capable to fit the best action possible for an encountered situation. On the other hand, because the context of supply chains often necessitates immediate reaction to changes, fast replanning and instant reply to customer, there is a need for responsiveness mostly provided through reactive behaviors. Therefore, hybrid agents exhibit high potential to support supply chain planning.

As presented earlier, the InteRRaP architecture provides an interesting approach able to react and deliberate when confronted with disturbances, using different capability levels.

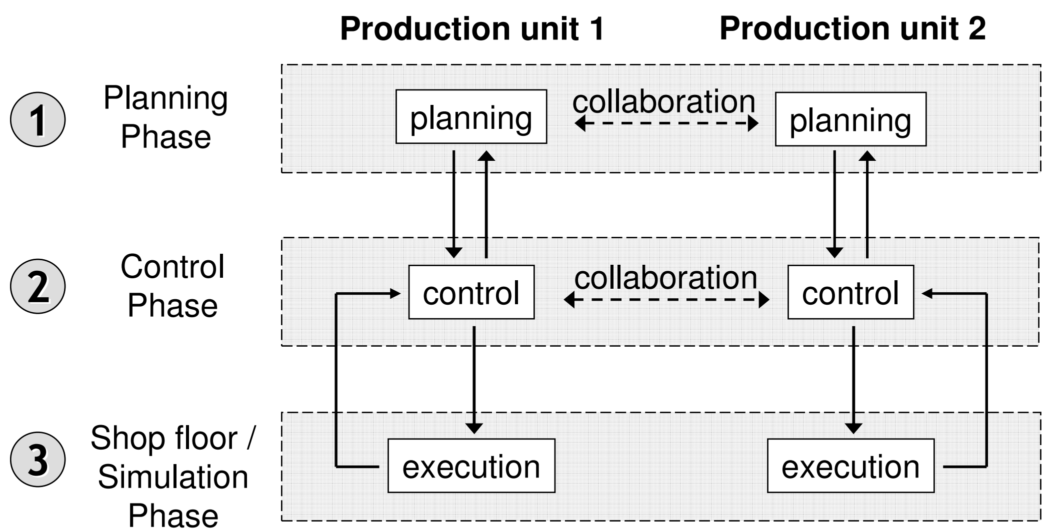
The agent can build action plans, depending if an event requires a reactive response, local planning or collaboration for planning. The Agent Building Shell (ABS) [17] is a collection of reusable software components and interfaces needed for any agent involved in a supply chain management system. The ABS is geared to handle perturbations caused by stochastic events in a supply chain. In this architecture, most of the efforts have been focused on defining communication and collaborative aspects. This is done through timely dissemination of information and coordinated revision of plans across the supply chain. The tri-base acquaintance model (3bA) [26] is a collaboration capable wrapper added to an agent. It provides the possibility of dealing with disturbances in a global perspective instead of resolving problems only in a local view. This is accomplished by using information about other agents without the need of a central facilitator. Theses authors present an example of applications in supply chains and they define the social knowledge needed to increase the efficiency of agents.

The literature proposes different agent-based solutions to help supply chain planning. Most of them use predetermined actions in order to react to specific disturbances. Although it can be sufficient in particular contexts, many situations require a sequence of tasks in line with agent own goals and partners' needs. Also, because more than one solution can by applied and a multitude of factors can have an impact on the best decision possible, the agent must know the different behaviors and when they can be applied. There is a need to specify these behaviors, how collaboration should be used and when the agent should use them.

3. Collaborative event management

To emphasize the importance of collaboration when dealing with disturbances in manufacturing supply chains, we first introduce the Collaborative Event Management (CEM) approach. This approach represents our general vision on how collaboration should be exploited to deal with disturbances (or events) within any manufacturing system in a supply chain. Due to interdependencies between business partners, there is a need to coordinate the production planning processes to solve problems resulting from disturbances in a timely and efficient way. In a CEM perspective, we represent manufacturing activities in three different phases (see Figure 1), which are the *planning* phase, the *control* phase and the *shop floor / simulation* phase, by presenting the interactions between two different production units.

Figure 1. Manufacturing phases in CEM



The planning phase (1) includes the creation of the initial plan by both production units individually. After initial planning, they collaborate and coordinate efforts to adjust plans, in order to reach the best profit. This collaboration between units is essential to insure production plan synchronization, in order to avoid delays and unfeasibly solutions. Capacity anticipation and scheduling anticipation can be used to help to build feasible plans and facilitate collaboration. Also, if a direct communication channel is not available or time is limited, it is possible to substitute collaboration by anticipation of partner's objectives. When completed, production plans are transmitted to the control phase (2), where validation and scheduling are performed. At this time, the control phase is used to dedicate resources to specific production tasks and make sure plans can be followed. The shop floor / simulation phase (3) uses scheduling plans transmitted to execute (or simulate) production. When a production unit is no longer able to follow the plan (because a perturbation of any kind), a local solution is looked for and deployed. When it is not possible to find such a solution, a feedback message is sent to the control phase, where collaboration can be used between production units to find a compromise in order to deal with the perturbation. Examples of compromises can be changes in delivery dates, changes in products and new production plans.

CEM puts collaboration at the heart of the planning activities, but leaves place for local correction when it is possible. With extended collaboration protocols and anticipation of the impacts of their decisions, it is possible to propose problem solving techniques to face unforeseen disturbances. Such an approach can smooth transitions in the supply chain, reducing safety stocks and lead times usually kept to cope with undesired impacts. CEM provides input to create agents with appropriate characteristics, especially when applied to an agent-based planning system, like the FOR@C experimental platform.

4. FOR@C Experimental platform

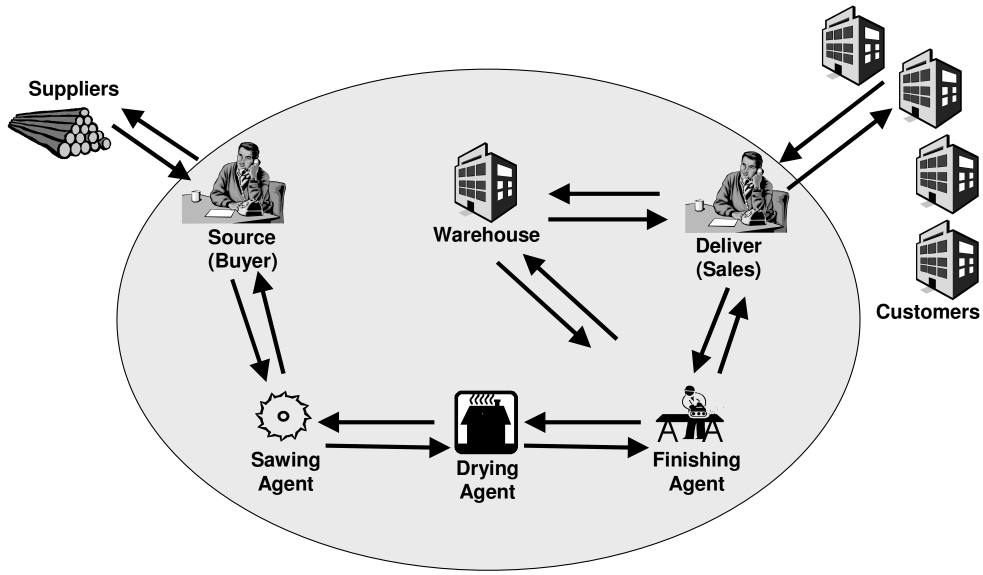
For many years, the planning processes in the North American lumber industry have not been questioned. Due to the highly heterogeneous nature of the resource (i.e. trees) and the inherent complexity of forecasting production throughput, the dominant thinking was to produce the maximum volume with the resource available (*push production*). Because of the commodity nature of the final product and the standards of sizes and grades, production is oriented towards large batches [19] to take advantage of economies of scale. This industry can be characterized by large inventories, low flexibility and low agility. The recent economic and international threats to the lumber industry have encouraged North American companies to rethink their planning processes. In order to compensate for the lack of control over the stochastic elements related to lumber production, an increase in the exchange of information between the different production centers is necessary, as is their ability to react quickly in a coordinated manner to changes. Also, in order better fulfill client needs and reduce missed sale opportunities, a *pull production* approach must be prioritized. In other words, instead of producing a maximum of expected high value products and offering them to the clients, the aim is to meet clients orders and to produce what is needed.

With the purpose of developing a new planning approach for the lumber supply chain, the FOR@C Research Consortium of the Université Laval (Quebec, Canada) has developed an experimental planning platform built on an agent-based architecture for advanced planning and scheduling (APS), with interaction mechanisms inspired from FIPA (Foundation for Intelligent Physical Agents) standards. This architecture combines agent-based technology with OR techniques to take advantage of the ability of agents to integrate distributed decision problems, and the ability of OR to solve complex decision problems [19]. Because of the distributed context of the supply chain and the use of agents, this experimental platform can be described more precisely as a distributed APS (or d-APS), where the first issue is to plan and coordinate all supply chain operations. This platform allows the different production centers to plan and correct deviance independently in line with their own needs, while maintaining feasibility and synchronization by collaborating with partners. By using a mix of pull and push production, each agent tries to first answer client's needs, and then complete the production plan with other products. By distributing planning decisions among specialized planning agents, using adapted optimization tools, the experimental platform increases agility in the supply chain. Also, the use of advanced conversation protocols between the agents insures the synchronization of production plans and a global feasibility for the supply chain.

4.1. Description of a planning unit

The agent-based architecture presented by FOR@C is based on the functional division of the planning domains, inspired by the SCOR model proposed by the Supply Chain Council [44]. Planning units divide activities between specialized production planning agents: a sawing agent, a drying agent and a finishing agent, since each of these planning problems are quite different in terms of the way the process and the set-up are conducted. Each of these agents is responsible for supporting the planning of his production center in terms of production output each day. Other agents are also part of the architecture, such as the deliver agent, source agent and warehouse agent. This paper focuses particularly on production planning agents. Figure 2 presents an example of a planning unit, including external exchanges with suppliers and customers.

Figure 2. Example of a planning unit from FOR@C experimental platform



The planning sequence used in a planning unit to plan the internal supply chain upon the receipt of a new demand plan (from outside the planning unit) is divided in two distinct planning phases: the infinite supply plan and the finite supply plan. During the first phase, the deliver agent receives a demand plan from one or many customers. These customers can be part of the same company or different companies. Upon reception, the deliver agent sends a demand plan to the warehouse agent to verify if the needed products are in stock. For non-available products, he sends a demand plan to finishing agent. Using this demand plan, along with resource constraints and lead times the finishing agent builds his plan considering infinite supply and transmits it to the drying agent. Again, using the demand plan, local constraint and considering infinite supply, the drying agent transmits his preferred plan to the sawing agent. This process continues until suppliers outside the planning unit receive the infinite demand plan. When suppliers answer the demand plans, the source agent receives a supply plan and starts a return loop. This represents the second phase of the planning process, the finite supply plan. The process is largely the same, however plans are built with finite supply, which is the information transmitted by the immediate supplier. For further information the reader is invited to read [19].

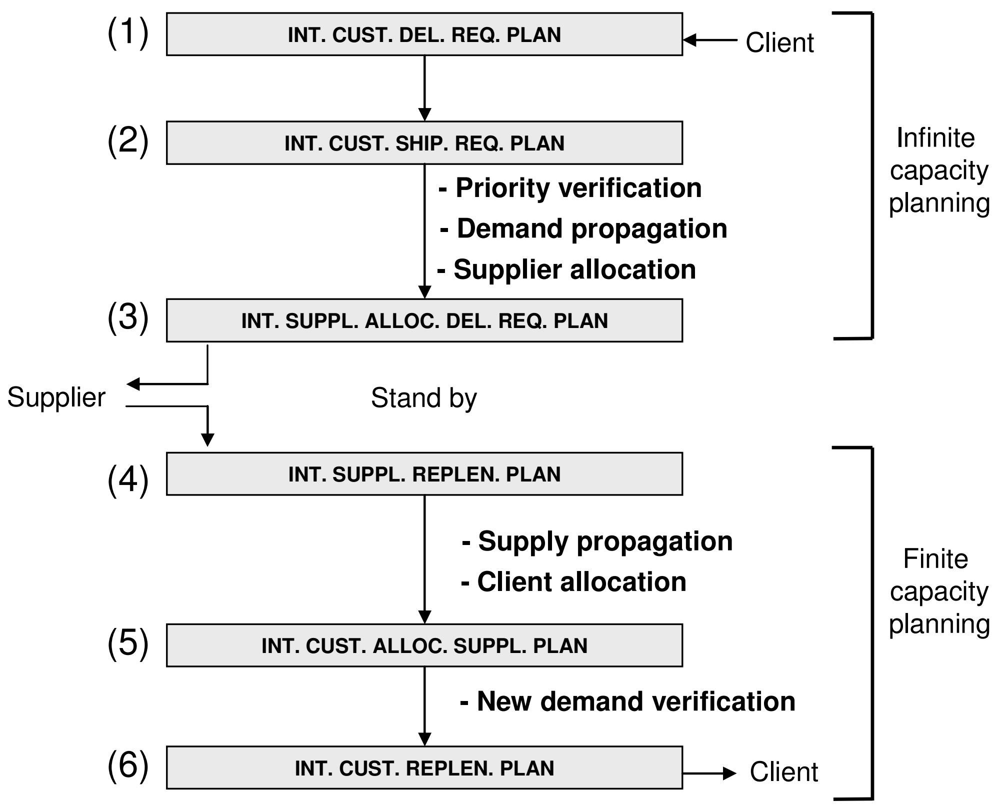
If an event occurs in the internal supply chain operations, any agent can initiate collaboration with his internal clients and suppliers by sending a revised demand or supply plan. This can be triggered by an agent who needs some products to fulfill inventory, lost production or new demand. This explains why agents are also responsible for continuously monitoring their environment and reacting to disturbances. Because of the interaction context, an agent’s environment is also made up of all messages received from other agents specifying a new or modified requirement plan, a new or modified replenishment plan, a contingency situation, or a high priority requirement to process.

4.2. Actions and task flows

Each planning agent has available objects (e.g. different types of supply or demand plans) which can be modified by local actions or actions from other agents. Actions are made possible by task flows, which are sequences of tasks, usually triggered by specific events. A planning agent’s standard task flow is the planning protocol (see Figure 3), which is triggered upon reception of a new demand plan from a client. Here, objects are represented by boxes and actions are presented in bold characters. This protocol is divided in two segments, where the first concerns the infinite production plan and the second the finite production plan.

Basically, when an agent receives a new demand plan, the object *Internal Customer Delivery Requirement Plan* (1) is modified. An offset corresponding to transportation delay is calculated and it modifies the *Internal Customer Shipping Requirement Plan* (2). From this last object, the agent starts different actions in a consecutive manner. First, a *Priority Verification* is pursued, which checks if the new requirement is urgent or if it is possible to wait until the next urgent requirement. Next, a new production plan is generated, using previous demand and new demand. This action is called *Demand Propagation*, because it is about translating demand from the client into demand to the supplier. In this first segment, because it is an infinite supply plan, the agent plans as if the supply was available and deliverable in time. In other words, it represents a wish that would optimize his production output. Then, a *Supplier Allocation* is conducted, which is the distribution of demand to the different suppliers. Optimization algorithms are performed in *Demand propagation* and *Supplier Allocation* actions, using linear programming, constraint programming and heuristics. This allocation modifies the *Internal Supplier Allocation Delivery Requirement Plan* (3), which is transmitted to concerned suppliers. At this time, the agent enters a standby period, waiting to get answers from suppliers.

Figure 3. Current planning protocol



The second segment starts upon the reception of supply plans from suppliers. This event modifies the *Internal Supplier Replenishment Plan* (4) object. From this supply plan, the agent starts a *Supply Propagation*, which is a new production plan built using real supply, called here finite supply. The production planned from this feasible plan is then allocated to the different clients (*Client Allocation).* This event triggers the modification of the

Internal Customer Allocation Supplier Plan (5) object. Then, verification of new orders is done to make sure no new demand arrived while processing, to avoid transmitting an already out-of-date supply plan to client. Finally, an *Internal Customer Replenishment Plan* (6) is modified and transmitted to clients. From there, if the client is another agent, the same planning protocol is started again.

4.3. Validation

The validation of these developments was carried out with the collaboration of a Canadian lumber company. Real data was used to test the performance of the agent-based APS. A supply chain configuration has been developed in order to address the planning of drying and finishing activities inside one plant. This configuration included different types of data, such as production processes, products, orders, on-hand inventory, selling prices, resource costs, forecasted supply, capacity and on-going work. This test covered 100 products, distributed over two kilns and one finishing line, over a rolling horizon of 6 weeks.

The first step of this validation was to model the drying and finishing processes with the partner's production manager. Loading patterns for kilns were known and available, but finishing processes were unknown. Work was done to define in details these processes, which resulted in 20 finishing processes and 89 drying processes. Customer order files and on-hand inventory data were extracted from the ERP system. The sales team provided the data on final product prices and resource costs. Each week, the partner's production manager sent us the execution plan, including supply from the sawing line, daily capacity of the finishing line and on-going work. The needed information was then translated into XML format and introduced into the experimental platform.

Approximately 80 exchange protocols, 100 tasks and 50 workflows were involved in the experimental planning platform. We then generated production and logistics plans, and presented these to the production manager for comments. This interactive validation phase allowed us to review and adjust the planning parameters and algorithms. By studying the real plan prepared by the manager, we were able to evaluate the performance of the platform in terms of number of late customer orders, production value, resource utilization, etc. These indicators, easily obtained by the platform, were precious to evaluate the performance of both plans and identify possible improvements. The validation process took approximately one year and many corrections have been made on the platform. Currently, plans generated by the platform offer considerable improvements when compared to plans prepared by the partner's production manager. For example, the actual planning times for drying and finishing operations have been reduced dramatically, going from one day (sometimes two days) to approximately 20 minutes.

The validation phase was crucial not only to verify the concept of the experimental platform and evaluate its performance, but also to collect information on how we could improve our concept and push it forward. The process to increase agility and synchronization of the supply chain has been started, especially when compared to general practices in the lumber industry. Nevertheless, much more can be achieved by, using the full potential offered by agent-based technology and advanced planning protocols. This leads us to the proposition of a new planning agent model.

5. Conceptual planning agent model

5.1. Enhancement of current planning agents

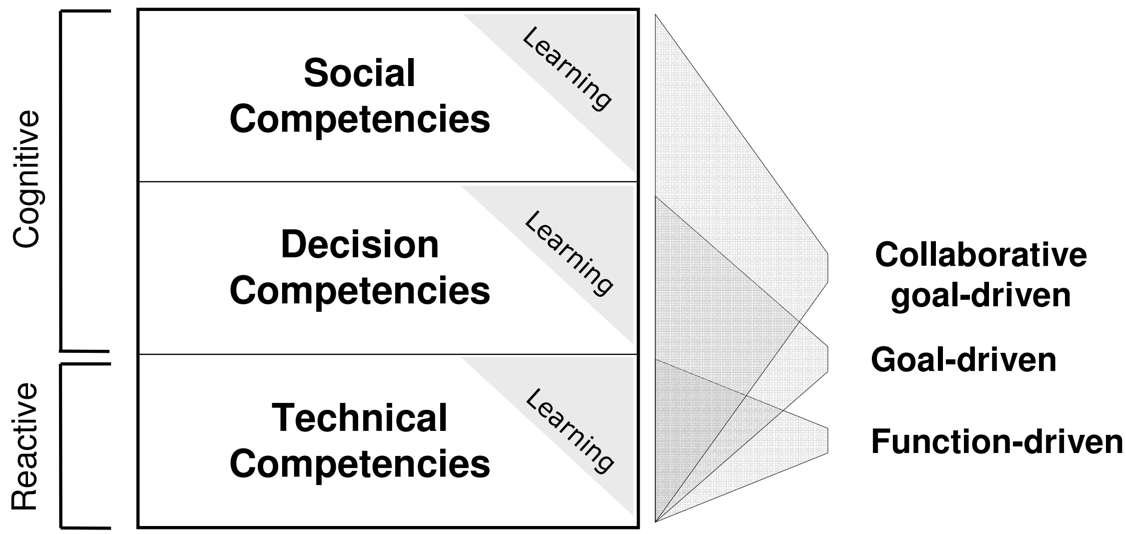
Agent-based planning systems, such as the experimental platform developed by the FOR@C consortium, represent a promising way to develop new planning systems in the supply chain, to improve global performance. Although much energy has been deployed to define and deploy the experimental platform, there are still possibilities for improvement, especially in the definition and design of the planning agent. The development of a new conceptual agent model aims to describe the characteristics needed to enhance agility and synchronization of current planning agents. Facing disturbances, these agents use reactive task flows, triggered by specific messages (from partners or disturbances). Our hypothesis is that agility and synchronization can be improved by using adapted behaviors (or strategies) depending on the situation and on the environmental context. To deploy agents with different behaviors, we must give the agent the ability to make choices and the capability to evaluate these choices following specific criteria or goals.

5.2. Model description

The agent conceptual model must present the competencies needed in order to show behaviors adapted to a dynamic supply chain context. Inspired by [23] and [36], we define competencies as the underlying attributes of an individual determining his capacity to successfully complete a task within a given environment. All competencies can be classified into three categories, which are attitudes, abilities and knowledge. Attitudes are the tendencies to act in a consistent way, following how an individual thinks and feels. Abilities are capabilities to perform specific tasks with the appropriate tools or techniques. Knowledge is defined here as the explicit understanding of information. In other words, the agent knows what the impact of the information is and how this information can be used. This has both having a direct impact on his behavior [30].

Integrating agent technology and OR tools, the conceptual model (Figure 4) is composed of three distinct layers, describing the different competencies required for supply chain planning. Here, the objective is to describe the basics capabilities in order to serve as a guideline for further developments, instead of a precise arrangement of functionalities. The model can be interpreted as a competency evolution model, depending on the agent use of the different competencies. This way, the agent can present a function-driven behavior, a goal-driven behavior or a collaborative goal-driven behavior.

Figure 4. Agent conceptual model



5.2.1. Competency layer descriptions

The bottom layer of the agent model is the *Technical Competency* layer. This decision layer includes tools, tasks and existing task flows, such as OR tools and algorithms, conversation protocols, negotiation protocols and queries. Attitude in this layer is related to minimizing effort while maximizing results, while abilities describe how to use tools, at the right time. The knowledge of the agent includes the possible tools and tasks, what are the parameterizations possible and the expected results. An agent that primarily uses this layer would show a *Function-Driven* behavior. Current agents deployed in the experimental platform exhibit such behavior. When they face a disturbance, they build a new production plan, send a new demand plan to suppliers and later, send a new supply plan to clients.

At this point, the agent is limited and a superior reasoning behavior could be achieved by giving new possibilities to agents, other than starting a global re-planning protocol. The agent would be greatly advantaged to have capabilities of analyzing the situation more deeply allowing it to make a clever decision. This is where the *Decision Competency* layer permits the evolution from reactive behavior to cognitive behavior. The agent attitude focuses now on maximizing progression toward agent own goals (or local goals). Abilities are about evaluating impacts of actions on local goals and how to choose the right action at the right time, following these impacts. Knowledge includes the explicit knowledge of local goals and impacts of decisions on himself. Geared toward the optimization of the goals it has been assigned to, the agent is primarily concerned by a set of performance metrics that represents what the systems designer has developed. In brief, the agent only knows the impacts of his decisions in terms of this set of metrics. Here, when a disturbance occurs, the agent has the capability to choose which task, task flow, optimization algorithms or complete plan could fit better, according to his own goals. The agent must have a representation of his goals and mechanisms to update and measure the achievement toward these internal goals. An agent oriented in the decision layer and technical layer would present a *Goal-driven* behavior. This additional competency clearly gives some advantage to the agent, but it is still unaware of the impact of his decisions on his partners, or on the supply chain. It needs a broader conception to be able to take decisions in the interest of the majority.

The *Social Competency* layer fills this gap by integrating the welfare of partners through collective goals. The agent is now aware of the impacts of his decisions on other agents and on the whole supply chain. While choosing actions to correct deviations from plan, the agent captures the entire potential of the network and is able to minimize impact on others. This layer includes mechanisms to obtain and update collective goals. Collective goals include other agent goals and network tactical goals (i.e. specific product, client selection, supplier selection). If the agent cannot have direct access to other agent goals or collective goals, he must be able to anticipate them. In this layer, the attitude is related to maximize progression toward partners' goals (or collective goals), abilities are about evaluating impacts of decisions partners and updating collective goals, and knowledge includes possible impacts of decisions on partners. With this competency layer, the agent can choose which task, task flow or plan responds best to his collective goals. Agents covering the three previous layers exhibit a *Collaborative goal-driven* behavior.

5.2.2. Learning competency

Embedded in each layer, the *learning competency* gives the agent the potential to increase his knowledge in each competency layer. A specific action or sequence of actions that demonstrated positive results in a situation could be learned and remembered for the next occurrence. The idea is to advance the articulation of the human decision-making process in our agent model. Various works have presented learning in agent-based systems as a way to improve the performance of manufacturing systems and supply chains. Shen et al. [40] present an interesting literature review on the subject and propose learning techniques adopted in the MetaMorph project. They distinguish learning from history (case-based reasoning) and learning from the future (by simulation). Alonso et al. [4] argue that learning is the most crucial characteristic of intelligent agent systems and present different learning perspectives and techniques. Although this subject is not detailed in this article, it will be studied in the near future, with the objective to be fully implemented in the experimental platform.

From this conceptual agent model presenting basic abilities, we need to clarify how these competencies can be used to increase agility and synchronization in a planning system. An extended agent model must be designed to implement an agent able to choose the correct planning action when confronted with a specific disturbance.

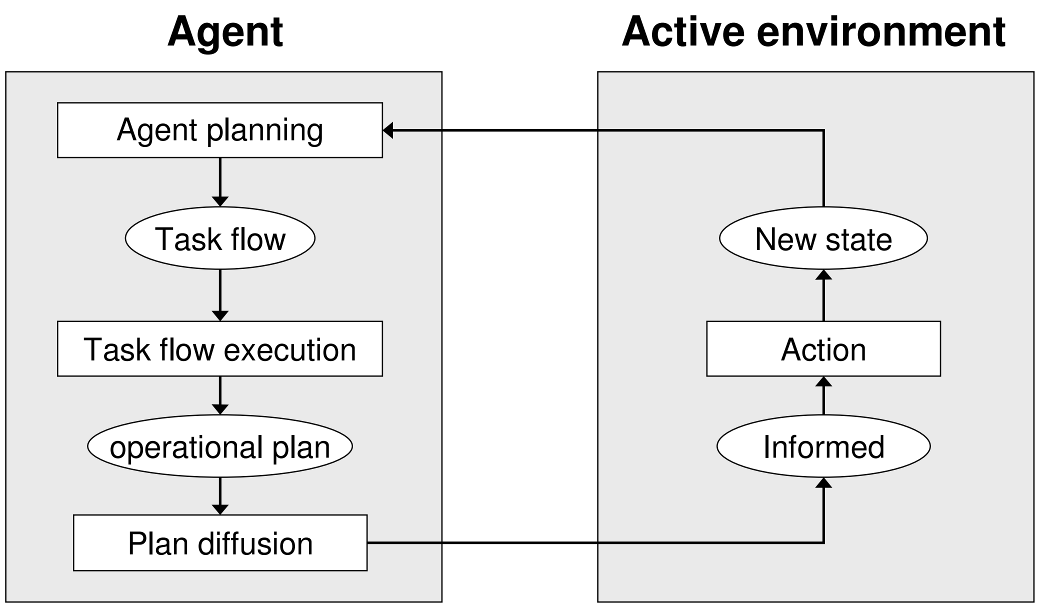
6. Multi-behavior agent model

The agent conceptual model presented in the above section describes some of the basic competencies necessary for a planning agent involved in dynamic supply chain planning. These competencies are quite general and there is a need to clarify how the agent can deal with all of them in a coherent global behavior. In this section, we present an agent meta-model and we describe the Multi-behavior agent model in detail, designed to offer different planning behaviors from which the agent can choose to solve the different planning problems.

6.1. Agent meta-model

The agent meta-model is a high level view of interactions between a planning agent and his active environment (Figure 5). Before actually building an operational plan, the agent must decide which task flow, or sequence of tasks, will be used, with the ultimate objective to build the best operational plan possible. Because the agent is not controlled by a central planning system, he is free to decide what he will perform, using his own preferences. In this meta-model activities are presented as boxes and results are ovals.

Figure 5. Agent meta-model



6.1.1. Meta-model description

From a new state in the environment, the agent first starts the *Agent Planning* phase. This phase is about planning the tasks of the agent. In other words, the agent deliberates to decide which attitude he should adopt and which ability he should used by building on his knowledge base, using different selection criteria, such as time available, chance of success of a particular task flow, source of the disturbance and private goals. This is where cognitive abilities are used since the agent has to choose every task, task flow and protocol he is going to perform to answer the disturbance. Its execution leads to the selection (or creation) of a task flow. The next phase is the *Task flow Execution*, which is chiefly the allocation of resources (machine, labor, etc.) to specific production tasks (for example, processing product A on machine 1 on day 4). Using a pre-determined algorithm, a production plan is built, creating demand plans for suppliers and supply plans for clients. Different techniques can be used here, from simple heuristics to complex constraint programming (CP) algorithms. The description of these production planning techniques is beyond the scope of this article, but the reader will find detailed information in [22]. The execution of a task flow leads to the creation of an operational plan. The *Plan Diffusion* phase distributes operation plans to every interested agent in the environment, including production staff related to the agent

On the active environment side, which includes all other agents, upon diffusion of operational plans, the environment becomes informed. *Actions* are performed in the environment, in order to respond to the change induced by the new operational plan. These actions lead to a new environment state. The planning agent is always watching the environment for a new state he can recognize. When it happens, the planning loop starts again. Even when no new state is noticed, the planning agent can perform a task flow. In these cases, the agent tries to increase his own performance instead of staying idle.

6.1.2. Agent reasoning

Agent planning is where reasoning is performed to choose between different planning alternatives. Researchers have proposed approaches to select the best task flow in a shop floor context, using case-based reasoning and heuristic search techniques [5]. Here, we use a utility evaluation method to compare different task flows, using specific parameters. We use four parameters:

1. Task flow (*TF*): sequence of tasks used to solve planning problem created by a disturbance;
2. Types of disturbances (*Dist*): new demand, new supply, execution problem, inventory error, etc.;
3. Available respond time (*Time*): transmitted by the client as a time limit inside which an answer must be transmitted (acceptance or refusal);

4. Intensity of disturbance (*Int*): percentage of changes since the last demand plan in term of quantity of products.

Utility can be defined as the degree of usefulness of a state to an agent [15, 35]. When alternative actions are possible to an agent, he chooses the action leading to the state with the highest utility. Utility theory is used to represent and reason about preferences, which are defined by the goals of the agent, as pre-defined by his designer. Examples of goals are maximizing demand satisfaction, minimizing number of late deliveries and maximizing profit. The agent calculates the utility of a task flow, reflecting the expected performance in term of progress towards goal completion. A way to implement the utility function is to use a learning database to store information about past performance of task flows confronted in disturbances and specific parameters.

From a list of task flows, the rational agent performs a reasoning function. This function checks the utility of each available task flow and selects the one with the highest utility. This statement can be represented as in equation 1.

$$\mathrm{Reasoning}(ListTF, Dist, Time, Int) := \underset{\forall\ TF \in ListTF}{\mathrm{argmax}}\ \mathrm{utility}(TF, Dist, Time, Int) \qquad (1)$$

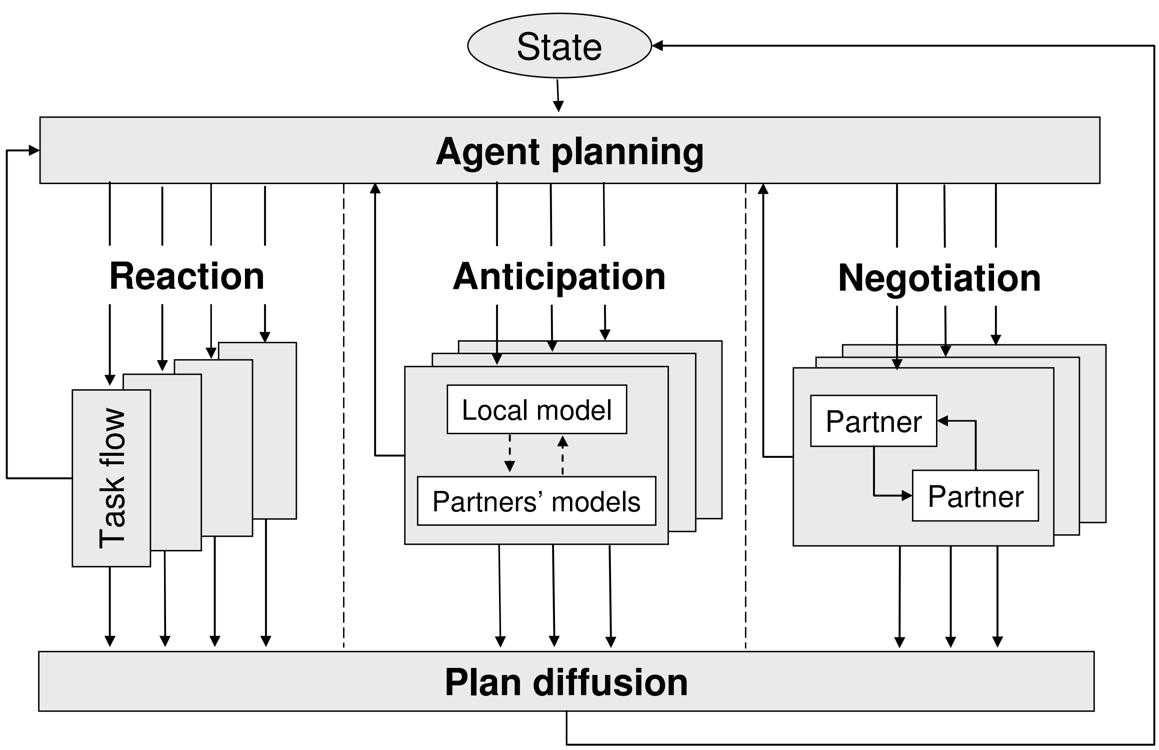
The *Agent Planning* phase also includes a reactive path to select a task flow. Disturbances in a specific context can identify special situations, such as no deliberation or utility calculation is needed. The responsiveness in these situations is improved since the agent does not have to evaluate every possible task flow. This is particularly interesting for situations where standard responses offer excellent results when confronted to specific disturbances. For example, if a planning agent receives a new demand plan requiring more than a 50% change when compared to the last demand plan received, it is strongly advised to run a complete infinite replanning task flow. These special situations must be chosen very carefully and can be adjusted when required.

Since the agent should show agility, he must be able to answer a disturbance with adapted actions and high responsiveness, choosing from a set of planning strategies and different production planning algorithms. The integration of all these possibilities raises the need for a more detailed agent model.

6.2. A new planning agent model

Using the agent meta-model as a basis, the Multi-behavior agent model (Figure 6) is an evolution of the concept. The model presents three basic behaviors to react to a new state in a planning context. Inspired by coordination mechanisms presented in [20], these planning behaviors are *Reaction*, *Anticipation* and *Negotiation*. Any task flow planned by the agent can be characterized by one of these behaviors.

Figure 6. Multi-behavior agent model



The *Reaction* behavior is about using task flows where no new information is collected during processing. The agent knows a certain number of task flows and can use one of them to respond to a disturbance. Different optimization algorithms and objective functions can be used, depending on the situation and on the available time. This behavior can be qualified as greedy, because the agent use only what is the best for him.

No knowledge about partners is use and there is no way to check if the proposed plan will satisfy the partner. These task flows are mainly used in well-known situations where no mutual adjustments are required. A large variety of task flows can be available, some of them taking much time but leading to optimal solutions, others finding acceptable (but not optimal) solutions in a very short period of time.

The *Anticipation* behavior is a planning strategy using partners' models in addition to the agent own local model. Basically, it is about integrating information about partners into his optimization model. Depending on the situation, emphasis can be placed on local or collective goals. Collaboration between planning partners through anticipation has been studied in hierarchical relation types [38] (also called upstream planning) and in a distributed context [37]. The partner model is used to plan production in a way to take into account his partners satisfaction, especially for production that has not been specifically asked for (push production). When no full disclosure is possible between partners (because of confidentiality needs), a partner's models are still used but represent a more approximated anticipation.

The *Negotiation* behavior describes task flows sending proposals to partners, in the form of alternative plans. When the agent is not able to respond to partner's needs, he can offer changes in delivery dates or alternative products. Following this, an iterative exchange of proposals is started, where both agents try to find a compromise. While both anticipation and reaction behaviors are non-convergent planning strategies, where the agent does not search for a compromise, the agent using a negotiation task flow is fully informed and tries to reach an agreement. These proposals can take the shape of new constraints, which can be used by partners to re-plan production and send a new demand plan. Before undertaking a negotiation protocol, the planning agent must determine a negotiation space, which specifies what parts of the plan can be changed. This way, the negotiation is narrowed and leads to a compromise faster. Negotiation between planning partners in supply chains has been studied in distributed relations [14, 43]. Dudek and Stadtler [14] propose a negotiation-based scheme between two supply chain partners, using a convergence mechanism based on exchange of local associated costs.

If there is still available time before diffusing the instructions to partners (*Plan Diffusion*), the agent can perform other task flows. In Figure 6, this is represented with feedback arrows. In this way, another reaction task flow can be executed, after undertaking a reaction task flow that showed low quality (e.g. high backorder, high inventory).

6.3. Advantages of the Multi-behavior agent

Compared to a purely reactive or deliberative agent, the Multi-behavior agent presents advantages similar to hybrid agents. For well-known situations, reaction task flows are used, but in situations where more information can be advantageous, the agent is able to demonstrate mutual adjustment capabilities, using anticipation or negotiation. The Multi-behavior agent proposed in this paper is a hybrid agent designed specifically to answer production planning problems, using different behaviors.

The main advantage is the possibility of adjusting the behavior according to external factors. For example, when a client sends a demand plan and requests an acceptance or a refusal in a short time frame, the agent is able to use his fastest respond, which is one of the reaction task flows. In this case, instead of entirely re-planning the production plan (that would take a certain amount of time), he would use on-hand inventory and try to satisfy the client's needs. In contrast, if a large amount of time is available, the agent would take time to send new demand plans to suppliers. This example is detailed in the next section.

Another advantage is the possibility for the agent to use collective goals in addition to local goals. Anticipation task flows use inputs from a partner's model in order to integrate both local and collective goals. Depending on the relative importance of these goals, a balanced solution can be reached. The possibility of anticipating collective goals when communication is not possible (or too long to achieve) represents an appreciable advantage, as better decisions can be taken with limited knowledge. Also, negotiation task flows use direct input from a partner. Instead of using an approximate model of collective goals, real local goals of partners are integrated in the final solution. These mutual adjustment approaches help planning agents find better solutions that would increase collective performance.

Although this description of advantages seems promising, it is still based on an untested agent model. A proof of concept is needed and performance measurements must be developed to claim any real advantages. This requires the implementation of the Multi-behavior agent architecture in a real-world supply chain context, where manufacturing activities are planned and confronted with stochastic disturbances.

7. Implementation in the lumber supply chain

In order to implement the Multi-behavior agent, it is necessary to develop different task flows in order to react efficiently to disturbances. Among the different disturbances present in the lumber industry (i.e. a major kiln breakdown, out of stock report, unmet harvest, etc.), we decided to first focus our efforts on a specific disturbance scenario, which is the reception of a new demand plan. A description of these task flows corresponding to the different planning behaviors is proposed.

7.1. Reception of a new demand plan

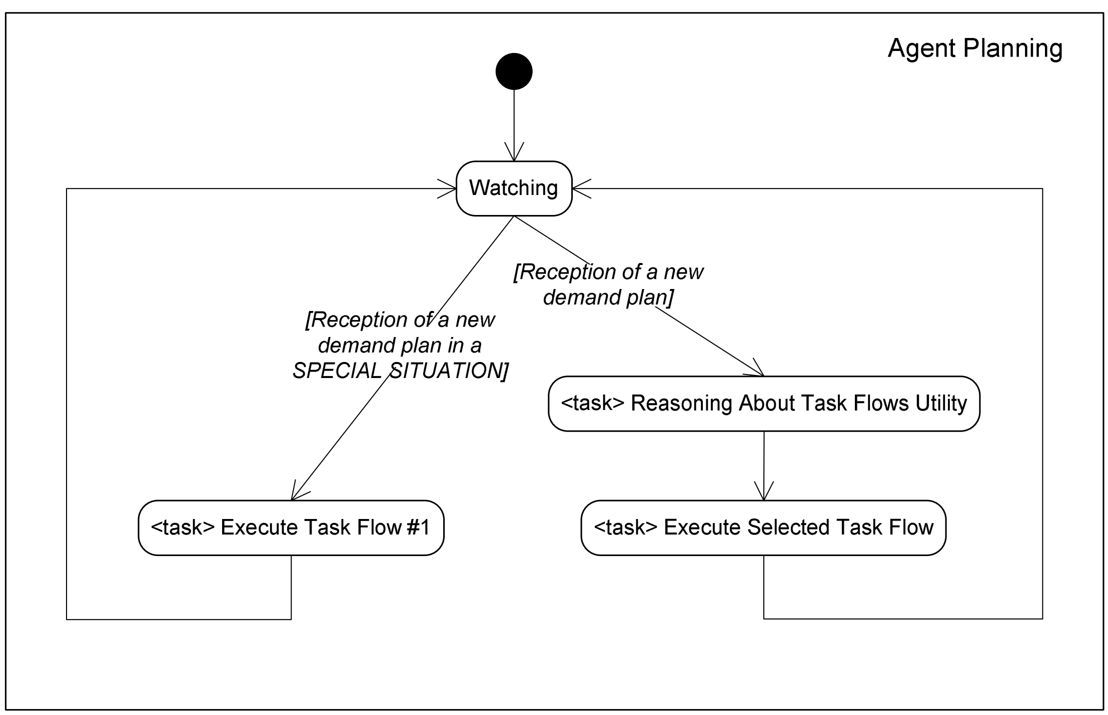
The scenario we retain here is the reception of a new demand plan by the planning agent from a client. A demand plan is formed of different product orders, requested for different dates. Following this new state, the planning agent must decide if the fulfillment of the new demand is possible or not. This decision is not as simple as it appears, mainly because of the time constraint. In many industries, such as the lumber industry, the available response time to give an answer is not unlimited and is sometimes in fact quite short. The main problem is giving an answer to the client in a way to maximize his satisfaction and to maximize local profit, inside the available time limit.

When the available respond time is large, the best option is to run a full production planning similar to the one previously presented in figure 3. Because available time is rarely that large, alternative planning processes must be available. Also, when the agent decides it is not possible to accept client demand, if there is still available time, he can still try to find another solution. Alternative plans can be proposed to clients, considering the modifications needed in his production plan, the resource availability and the delivery dates requested. This is where the *Reaction*, *Anticipation* and *Negotiation* task flows are involved.

7.1.1. Agent planning reasoning flow

In the agent planning phase, the agent must select the best task flow, using either a reactive or a deliberative path. Applied to the new demand plan scenario, the reasoning flow can be represented as in Figure 7. Upon reception of a new demand plan, the agent must first check if it corresponds to a special situation. If it is the case, it triggers a specific task flow (for example, task flow #1). Otherwise, a reasoning task is performed, in order to evaluate the utility of the available task flow and select the one with the highest utility. The selected task flow is then executed.

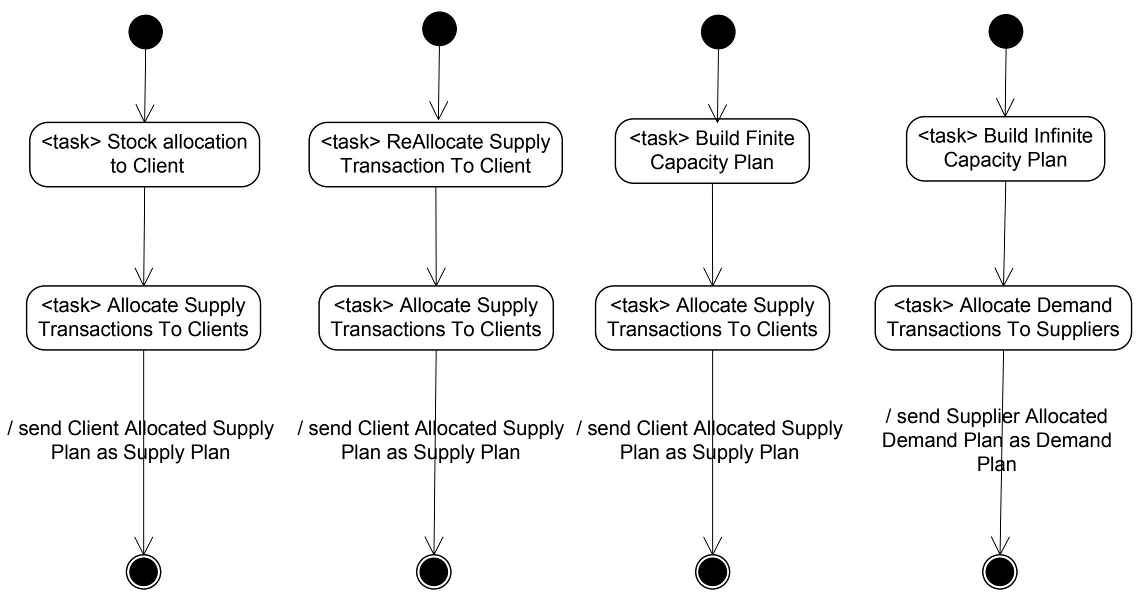
Figure 7. Example of an Agent Planning reasoning flow



7.1.2. Reaction task flow

All *Reaction* task flows require performing a production planning task. This can require building a new production plan or adapting the current one. The infinite production planning previously presented (first segment of Figure 3) is an example of a Reaction task flow triggered by reception of a new demand plan. Other Reaction task flows can be used in order to answer different needs. For example, instead of running a full infinite production planning, which takes quite a long time because suppliers have to send demand plans to their own suppliers, the agent can try to fill the new needs by re-allocating available stocks. He can also re-allocate supplies previously reserved for other clients, without involving any new delay. Another option is to re-run a finite production planning task without asking for new supplies. Figure 8 presents an example of *Reaction* task flows.

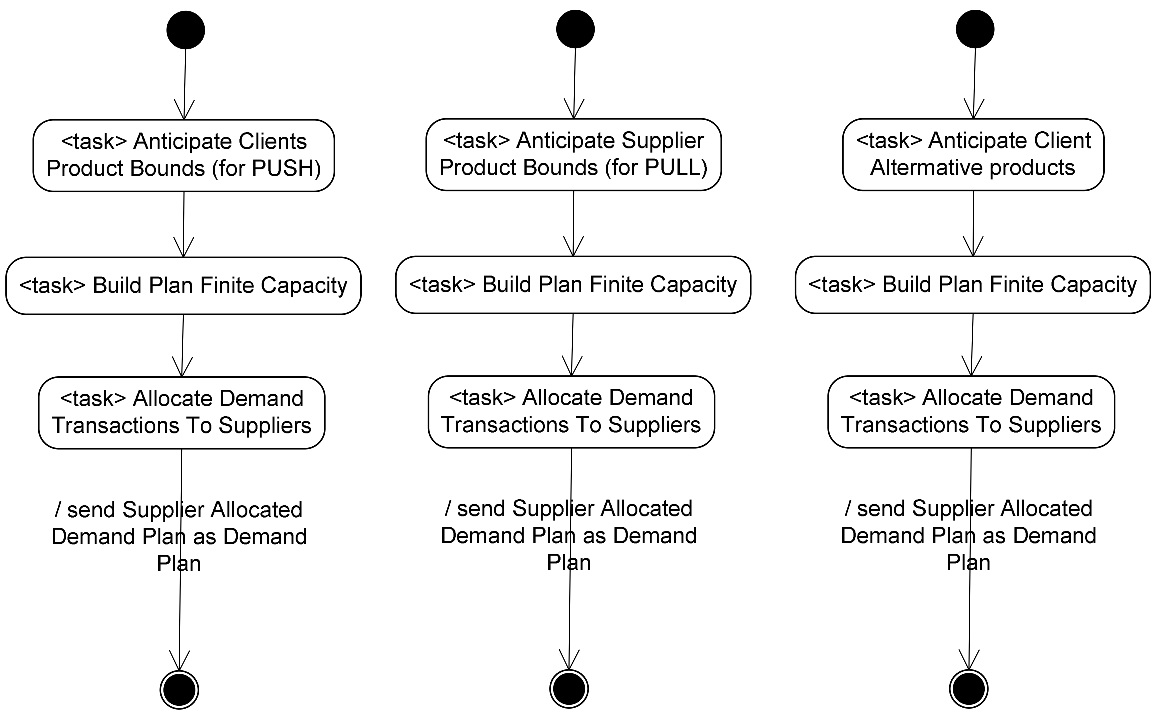
Figure 8. Examples of Reaction task flows



7.1.3. Anticipation task flow

As introduced previously, *Anticipation* task flows are about using a partner's model in order to guide production planning. When the planning agent receives a new demand plan, he can anticipate information from his client and supplier, in order to plan his production the best way possible. For example, because the experimental planning platform proposes a mix of pull and push production (pull production is required by the client, while push production is offered by the supplier but not required by the client), the planning agent plans push production with the remaining production capacity. In this case, it is possible to anticipate client's needs for push production, following a model of his inventory. Also, the agent can anticipate his supplier's production capacity to create a demand plan (in pull production) in line with the capacity limit. Another example is the offer of alternative products. When the agent is not able to fulfill the demand plan from his client, he can offer substitute products that the client would possibly accept (for example sending higher grade products). Figure 9 presents examples of such task flows.

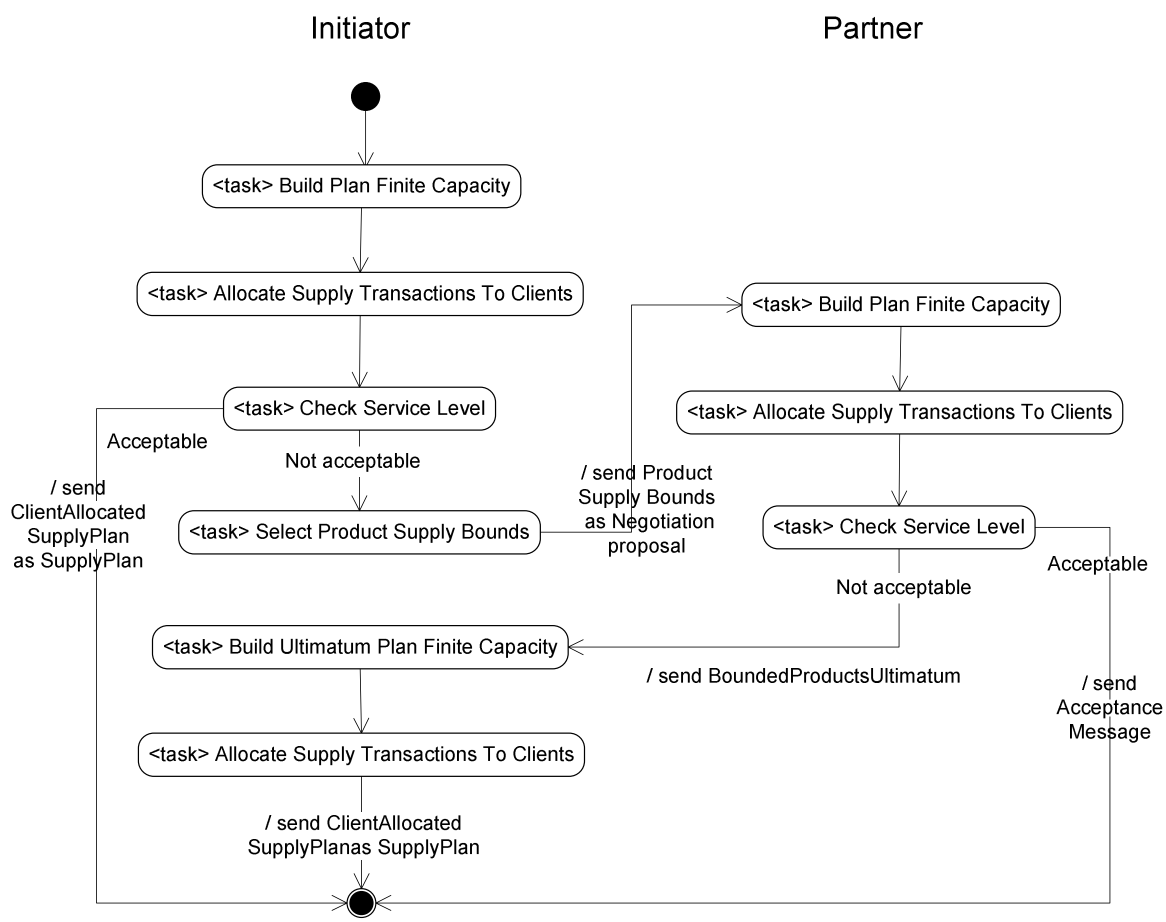
Figure 9. Examples of Anticipation task flows



7.1.4. Negotiation task flow

The last behavior concerns direct communication with partners, in the form of a negotiation protocol. This behavior is preferred when a partner's model is incomplete or when the partner's input is needed. It is particularly useful in situations where the planning agent is not able to offer a positive answer to the client, but time is available to transmit production limits to the client. For example, if the agent is not able to produce what is asked in the demand plan, he can transmit a new set of boundaries, in order to give to the client new data to rerun a new demand plan. These boundaries can take the form of limits on specific products (for example, maximum of 2000 million board feet of pine on November 16). Upon reception of these new boundaries, the partner builds a new production plan in finite capacity using these new boundaries. After checking if these changes are acceptable or not for him, the partner will send an acceptance message or a new set of ultimatum boundaries on the same product. These ultimatum boundaries represent a final offer the initiator. If these last boundaries are not accepted, the deal is over and the demand plan is erased. Figure 10 presents an example of a *Negotiation* task flow.

Figure 10. Example of a Negotiation task flow



7.1.5. Scenario progress

Based on this demand plan scenario and on the task flows presented (Figures 8, 9 & 10), an example of the scenario progress can be described. In this case, the client gives a delay of one hour to accept or refuse his demand plan, and this new plan involves a total increase of 20% of the demand for this period on time. According to the Multi-behavior agent model (Figure 6), the first action taken by the planning agent is to plan what to do by evaluating which task flow is the most appropriate. With this short delay, the agent clearly does not have time for negotiation or infinite production planning (both requiring waiting time). The remaining options need to be evaluated using his utility function. The planning agent can try to fit the new demand plan into the current production plan by building a new finite production plan. Another possibility is to reassign stocks or on-line productions promised to another client to accommodate the new client. From the utility evaluation of these options, a decision is taken, such as building a finite production plan. The selected task flow is executed, resulting in the creation of a new supply plan for his client. If the client is completely satisfied, the demand plan is transmitted. If the client is not satisfied (meaning the planning agent is not able to deliver every product in time) and if there is still available time, another task flow can be performed. The agent reruns a new utility evaluation and tries to find a new solution. In this example, by reassigning supplies, the agent can find a way to fit the client's demand plan in. The agent updates his production plan and sends an acceptance message to his client.

This example demonstrates the planning possibilities of the Multi-behavior agent model and the advantages of using such an agent model in a supply chain planning system. By adapting his behavior to the situation, using reaction, anticipation or negotiation protocols, the agent can react promptly and use the best strategy for each different situation.

7.2. Simulation plan

In order to prove the concept of the Multi-behavior agent and test its performance, implementation and simulation must be undertaken on the FOR@C experimental platform. Implementation will be gradual and behaviors will be developed successively. The first implementation will be the *Reaction* work flows. This step includes the development of the agent planning capability, including the utility evaluation function. All reaction task flows must be created and tested independently. Also, performance statistics must be compiled for each of these task flows in order to give information about the chance of success of executing a task flow in a specific situation. At this stage, it will be possible to simulate all *Reaction* task flows on the experimental platform, by designing a supply chain made of *Reaction* planning agents (agents using only *Reaction* task flows). Performance tests will be possible by comparing key performance factors (i.e. resource use, rapidity of response, etc.) of this supply chain with the current implementation.

The second implementation will involve the *Anticipation* behavior. This includes the introduction of partners' models, in order to give the planning agent information about clients and suppliers. Here, testing will be possible by comparing a supply chain made with agents using exclusively *Anticipation* task flows with a *Reaction* supply chain and the current implementation. Comparison can also be made concerning priority given to a partner's model or on the local model. This will be used to decide whether to follow local or collective goals first.

The final implementation will be the *Negotiation* behavior. All *Negotiation* protocols will be developed, including convergence mechanisms to ensure reaching compromises. Again, comparison of performances will be possible with previously the mentioned supply chain configurations.

8. Conclusion

Supply chain planning agent models which use the advantage of reactivity, utility evaluation, anticipation and negotiation, such as the Multi-behavior agent, can be a powerful tool to reach appreciated gains when implemented in a distributed planning system such as the FOR@C experimental platform. Following the conceptualization of the required intelligent behaviors and their implementation, future work is needed. For example, we intend to test different agent configurations in real-world planning situations to determine the different situations where specific behaviors react well and those where they react badly. In a different perspective, it will be of great interest to increase research efforts on the learning competency, with both its implications and impacts. A Multi-behavior agent geared with learning abilities would be able to update its utility functions to modify its preference for an action which gave good results in the past. This is highly promising and should lead to an even more agile and performing supply chain.

Acknowledgement

This work was funded by the FOR@C Research Consortium and the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

[1] C. Abid, S. D'Amours & B. Montreuil, "Collaborative order management in distributed manufacturing", International Journal of Production Research, vol. 42, no. 2, 2004, pp. 283-302

[2] R.J. Abumaizar & J.A. Svestka, "Rescheduling job shops under random disruptions", International Journal of Production Research, vol. 35, no. 7, 1997, pp. 2065-2082

[3] M.S. Akturk & E. Gorgulu, "Match-up scheduling under a machine breakdown", European Journal of Operational Research, vol. 112, 1999, pp. 81-97

[4] E. Alonso, M. D'Inverno, D. Kudenko, M. Luck & J. Noble, "Learning in multi-agent systems", The Knowledge Engineering Review, vol. 16, no.3, 2001, pp. 277-284

[5] H. Aytug, M.A. Lawley, K. McKay, S. Mohan & R. Uzsoy "Executing production schedules in the face of uncertainties: A review and some future directions", European Journal of Operational Research, vol. 161, 2005, pp. 86-110

[6] A. Azevedo, C. Toscano, J.P. Sousa, & A.L. Soares, "An advanced agent-based order planning system for dynamic networked enterprises", Production Planning & Control, Vol. 15, No.2, 2004, pp. 133-144

[7] A. Barua, N. Raghavan, A. Upasani & R. Uzsoy, "Implementing global factory schedules in the face of stochastic disruptions", International Journal of Production Research, vol. 43, no. 4, 2005, pp. 793-818

[8] R.A. Brooks, "A robust layered control system for a mobile robot", IEEE Journal of Robotics and Automation, vol. 2, no. 1, 1986, pp. 14-23

[9] S. Bussmann, N.R. Jennings & M. Wooldridge, Multiagent Systems for manufacturing Control: A Design Methodology, 2004, Springer-Verlag

[10] M. Caridi & S. Cavalieri, "Multi-agent systems in production planning and control: an overview", Production Planning & Control, vol. 15, no. 2, 2004, pp. 106-118

[11] L. Cloutier, J.M. Frayret, S. D'Amours, B. Espinasse & B. Montreuil, "A commitment-oriented framework for networked manufacturing co-ordination", International Journal of Computer Integrated Manufacturing, vol. 14, no. 6, 2001, pp. 522-534

[12] R.L. Daniels & P. Kouvelis, "Robust Scheduling to Hedge against Processing Time Uncertainty in Single-Stage Production", Management Science, vol. 41, no. 2, 1995, pp. 363-376

[13] T. Davis, “Effective Supply Chain Management”, Sloan Management Review, summer 1993, pp. 35-46
[14] G. Dudek & H. Stadtler, “Negotiation-based collaborative planning between supply chains partners”, European Journal of Operational Research, vol. 163, 2005, pp. 668-687
[15] P.C. Fishburn, Utility Theory for Decision Making, 1970, John Wiley and Sons
[16] M. Fletcher, R.W. Brennan, D.H. Norrie & M. Fleetwood, “Reconfiguring processes in a holonic sawmill”, 2001 IEEE International Conference on Systems, Man, and Cybernetics, vol.1, 2001, pp. 158-163
[17] M.S. Fox, M. Barbuceanu & R. Teigen, “Agent-Oriented Supply-Chain Management”, International Journal of Flexible Manufacturing System, vol. 12, 2000, pp. 165-188
[18] J.M. Frayret, A conceptual framework to operate collaborative manufacturing networks, Doctoral thesis, Université Laval, 2002
[19] J.M. Frayret, S. D’Amours, A. Rousseau & S. Harvey, Agent-Based Supply Chain Planning In The Forest Products Industry, Working Paper Dt-2005-Jmf-1, Centor, Université Laval, 2005
[20] J.M. Frayret, S. D’Amours & B. Montreuil, “Coordination and Control in Distributed and Agent-Based Manufacturing Systems”, Production Planning & Control, vol. 15, no. 1, 2004, pp. 42-54
[21] G. Gardner, Frames of mind: the theory of multiple intelligences, 1983, New York: Basic Books
[22] J. Gaudreault, J.M. Frayret, A. Rousseau, S. D'Amours, “Integrated planning and scheduling in a divergent production system with co production: a real-time perspective”, Centre CENTOR, Université Laval, Working paper DT-2006-JMF-1, 2006
[23] T. Hoffmann, “The meanings of competency”, Journal of European Industrial Training, vol. 23, no. 6, 1999, pp. 275-285
[24] R. Kerr & E. Szelke, Artificial Intelligence in Reactive Scheduling, 1995, Chapman & Hall
[25] H.L. Lee, V. Padmanabhan, S. Whang, “The Bullwhip Effect in Supply Chains”, Sloan Management Review, Spring 1997, pp. 93-102
[26] V. Marik, M. Pechoucek, O. Stepankova, “Social Knowledge in Multi-agent Systems”, Multi-Agent Systems and Applications, ACAI 2001, LNAI 2086, 2001, Springer, pp. 211-245
[27] F.P. Maturana, W. Shen & D.H. Norrie, “Metamorph: An Adaptive Agent-Based Architecture for Intelligent Manufacturing”, International Journal of Production Research, vol. 37, no. 10, 1999, pp. 2159-2174
[28] B. Montreuil, J.M. Frayret & S. D’Amours, “A Strategic Framework for Networked Manufacturing”, Computers in Industry, vol. 42, 2000, pp. 299-317
[29] J.P. Muller, “The Design of Intelligent Agents: A Layered Approach”, Lecture Notes in Artificial Intelligence, LNAI 1177, 1996, Springer, Berlin
[30] A. Newell, “The Knowledge Level”, Artificial Intelligence, vol. 18, 1982, pp. 87-127
[31] H.V.D Parunak, Practical and Industrial Applications of Agent-based Systems, 1998, Industrial Technology Institute
[32] M. Pechoucek, J. Vokrinek & P. Becvar, “ExPlanTech: Multiagent Support for Manufacturing Decision Making”, IEEE Intelligent Systems, vol. 20, no. 1, 2005, pp. 67-74
[33] L. Pryor & G. Collins, “Planning for Contingencies: A Decision-based Approach”, Journal of Artificial Intelligence Research, vol. 4, 1996, pp. 287-339
[34] A.S. Rao & M.P. Georgeff, “An abstract architecture for rational agents”, In Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning (KR’92), 1992, Boston, MA, pp. 439-449
[35] S. Russel & P. Norvig, Artificial Intelligence: A Modern Approach, Second Edition, 2003, Prentice Hall, New Jersey
[36] F. Salvador & C. Forza, “Individual competences and training needs for mass customization”, In 3rd Interdisciplinary World Congress on Mass Customization and Customization, 2005, Hong Kong
[37] C. Schneeweiss, Distributed Decision Making, Second Edition, 2003, Springer-Verlag
[38] C. Schneeweiss & K. Zimmer, “Hierarchical coordination mechanisms within the supply chain”, European Journal of Operational Research, vol. 153, 2004, pp. 687-703
[39] E. Szelke & G. Markus, “A blackboard based perspective of reactive scheduling”, in Artificial Intelligence in Reactive Scheduling, 1995, Chapman & Hall
[40] W. Shen, F. Maturana & D.H. Norrie, “Enhancing the performance of an agent-based manufacturing system through learning and forecasting”, Journal of Intelligent Manufacturing, vol.11, no. 4, 2000, pp. 365-380

[41] W. Shen & D.H. Norrie, “Agent-Based Systems for Intelligent Manufacturing: A State-of-the-Art Survey”, Knowledge Information Systems, vol. 1, no. 2, 1999, pp. 129-156
[42] W. Shen, D.H. Norrie & J.P. Barthès, Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing, 2001, Taylor & Francis, London
[43] H. Stadtler, “Supply chain management and advanced planning – basics, overview and challenges”, European Journal of Operational Research, vol. 163, 2005, pp. 575-588
[44] S. Stephens, "The Supply Chain Council and the Supply Chain Operations Reference (SCOR) model: Integrating processes, performance measurements, technology and best practice", Logistics Spectrum, vol. 34, No.3, 2000, pp. 16-18
[45] T.J. Strader, F.R. Lin & M.J. Shaw, “Simulation of Order Fulfillment in Divergent Assembly Supply Chains”, Journal of Artificial Societies and Social Simulation, 1998, vol. 1, no. 2
[46] G. Weiss, Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, 2003, The MIT Press, Cambridge, Massachusetts