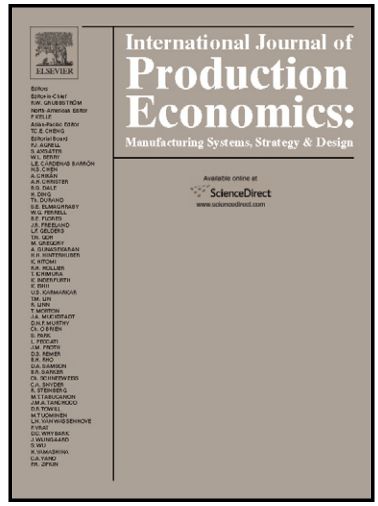
Author's Accepted Manuscript



www.elsevier.com/locate/ijpe

An autonomous multi-agent approach to supply chain event management

Lorena A. Bearzotti, Enrique Salomone, Omar J. Chiotti

PII: S0925-5273(11)00377-X
DOI: doi:10.1016/j.ijpe.2011.08.023
Reference: PROECO 4852

To appear in: *International Journal of Production Economics*

Received date: 11 January 2011
Revised date: 17 August 2011
Accepted date: 24 August 2011

An Autonomous Multi-Agent Approach to Supply Chain Event Management

Lorena A. Bearzotti, Enrique Salomone[1], and Omar J. Chiotti[2]*
[*]Universidad Nacional Andrés Bello, Campus Los Castaños 7 Norte 1348, Viña del Mar, Chile
[1]INGAR – CONICET, Avellaneda 3657, Santa Fe Argentina, 3000
[2]GIDSTAD – UTN FRSF, Lavaise 610, Santa Fe, Argentina, 3000
lo.bearzotti@uandresbello.edu
{salomone, chiotti}@santafe-conicet.gov.ar

Abstract— *Organizations have made significant effort to implement software for planning and scheduling, but disruptive events management is still a problem to be solved. Because of a disruptive event can affect the overall performance of the supply chain, SCEM (Supply Chain Event Management) systems presenting different automation levels such as monitoring, alarm, and decision support have been proposed. However, the management of disruptive events, taking into account the distributed nature of the supply chain, the members´ autonomy, and the ability to exert corrective control actions, has been identified as a problem that requires further research. This work presents an agent-based approach for the SCEM problem, which can perform autonomous corrective control actions to minimize the effect of deviations in the plan that is currently being executed. These control actions consist of a distribution of the variation between supply chain members, using the plan's slack in a collaborative way. An innovative feature of this approach is its focus on resources, which are affected by disruptive events in a direct way. Based on this approach, a SCEM system is designed as a net of control points defined on resources connected through supply process orders. Two novel aspects are the distributed collaborative inter-organizational architecture of the SCEM system and a Double Contract Net Protocol. This protocol allows a set of resource-representing agents to interact through an agent, representing a supply process order as a mediator. An application to a case study of the Multi-Agent SCEM system implemented with JADE is provided.*

Keywords – supply chain, event management, multi-agent system, autonomous behavior.

*Corresponding author
* Lorena Bearzotti
Universidad Nacional Andrés Bello, Campus Los Castaños 7 Norte 1348, Viña del Mar, Chile
Phone: +56-32-2845000
Fax: +56-32-2845284
Email: lo.bearzotti@uandresbello.edu
Enrique Salomone: salomone@santafe-conicet.gov.ar
Omar Chiotti: chiotti@santafe-conicet.gov.ar

An Autonomous Multi-Agent Approach to Supply Chain Event Management

1. Introduction

In a Supply Chain (SC) operational problems caused by disruptive events during process execution occur daily affecting the plans of both the organization where they are produced and the other members of the SC (Swaminathan et al, 1998; Chopra and Meindel, 2001; Christopher, 2005; Shapiro, 2000; Simchi-Levi et al, 1999). In this work, a *disruptive event* is defined as a significant change in planned values of the availability of the resources that an order requires to be executed. Examples are: equipment breakdown and breakage of materials (Meydanoglu, 2009; Masing, 2003; Knickle and Kenneler, 2002).

The occurrence of disruptive events is a well-known fact in the planning task. Therefore, planning systems generate plans including slacks to be robust and flexible. In this way, the plan can be adjusted to conditions occurring during implementation (van Landeghem and van Maele, 2002; Adhitya et al, 2007; Wang and Lin, 2009; Bui et al, 2009; Liu and Min, 2008).

Supply Chain Event Management (SCEM) is defined as the business process where disruptive events are recognized in a timely fashion, actions are quickly triggered, material and information flows are adjusted, and key employees are notified. The goal is to enable the SC to respond to disruptive events by minimizing their impact, thus avoiding the need of re-planning. This implies assessing, monitoring and evaluating disruptive events within and across companies and initiating consequent actions. To support this business process a new generation of information systems, known as SCEM Systems has emerged (Masing, 2003; Zimmermann, 2006).

SCEM systems can present different automation levels, and thus they can be classified in the following types: *monitoring systems*, planned as an extension of traditional tracking and tracing systems (Szirbik et al, 2000; Kärkkäinen et al, 2003); *alarm systems*, which can detect deviations in the plan and notify the key employees (Hoffmann et al, 1999; Teuteberg and Schreber, 2005; Zimmermann, 2006): *decision support system*, which can detect deviation and propose to the human decision-maker solutions that minimize the disturbance impact on the system (Speyerer and Zeller, 2004; Adhitya et al, 2007; Cauvin et al, 2009); *autonomous corrective systems*, able to detect a disruptive event, look for a solution and implement it if there is any.

This work presents an agent-based architecture for a SCEM system. This proposal introduces two novel aspects not addressed in previous works. Firstly, the system is conceived as a distributed collaborative inter-organizational information system in response to the need for an approach for managing disruptive events by taking into account the distributed nature of SC while preserving members´ autonomy (Cauvin et al, 2009). Secondly, mechanisms are provided for the system to perform autonomous corrective control actions. This aspect is included in response to a requirement not fulfilled by the existing proposals, which are mainly focused on addressing tasks of monitoring, capturing, and communicating disruptive events. The ability to exert corrective control actions has been identified as a barely explored area (Zimmermann, 2006; Pereira, 2009). In this approach, control actions use the slack in the plan to search for a solution to mitigate the disruptive effect. Another novel approach proposed in this paper is the Double Contract Net Protocol. This consists of a protocol a coordinator agent starts by executing a Contract Net protocol with a mediator agent, which in turn starts many Contract Net protocols with as many responder agents as necessary.

In the following section, related works are presented. The proposal is developed in Section 3, and in Section 4, an illustrative example is shown. Conclusions and future works are presented in Section 5.

2. Related Works

In recent years the problem of systematically identifying and correcting disruptions and malfunctions in operational supply-chain processes has been studied both in industry and academia. Most relevant approaches to this work are discussed below.

Hoffmann et al (1999) presented ECTL-Monitor, an agent-based system designed to be embedded in the company's Internet portal in order to provide customers with elements to track and trace their orders and set up notification services. These features are characteristic of an *alarm system*. The system is not provided with mechanisms to perform autonomous corrective actions when a disruptive event occurs, and it is not conceived as a collaborative inter-organizational information system.

Szirbik et al (2000) presented PROVE, a prototype based on mobile-agents with focus into bringing support to negotiation among enterprises for building and maintaining a virtual enterprise (target cluster). A disruptive event takes place when an enterprise leaves the cluster, and in this case, the approach provides support to find an enterprise that could assume the commitment. This is a semi-structured decision process that can not be fully automated, and the human intervention is necessary. These functionalities are characteristic of a *decision support system*. PROVE also provides support for tracking and monitoring of orders; typical functionalities of a *monitoring system*. Whilst it was designed as an inter-organizational information system, it is not provided with mechanisms to perform autonomous corrective actions when disruptive events, other than the abandonment of the cluster by an enterprise, occur.

Kärkkäinen et al. (2003) presented DIALOG, an agent-based system that shares data to facilitate order tracking by offering information on its state. This functionality is characteristic of a *monitoring system*. It was designed as an inter-organizational information system, but abilities to perform autonomous and collaborative corrective actions are not provided.

Speyerer and Zeller (2004) presented FORWIN, a web-based prototype system that provides support for alarm and monitoring through its ability to detect performance deviation, identify deviation causes, and propose potential mitigation measures. Symptoms of performance problems are gathered from supervision of pre-defined metrics. Its functionalities are characteristic of a *decision support system* and it was conceived as an inter-organizational information system. But it is not provided with abilities to perform autonomous and collaborative corrective actions at the level of orders and resources associated to an operational plan under execution when a disruptive event occurs. It is focused on the high-level management of the supply chain, leaving the monitoring and control of order execution processes out of its scope.

Teuteberg and Schreber (2005) presented CoS.MA, a peer-to-peer system designed to integrate data from single members so that all members have a visualization of pertinent data. Order tracking and tracing in the SC is supported by using Auto-ID and mobile technologies. Its functionalities are characteristic of an *alarm system*. It was designed as an inter-organizational information system, but not to perform autonomous and collaborative corrective actions to mitigate the effects of disruptive events.

Bodendorf and Zimmermann (2005) and Zimmermann (2006) presented PAMAS, an agent-based SCEM system based on monitoring orders as they move in the SC detecting when a disruptive event affects an order. It uses adaptive order profiles to identify orders to be monitored because some of their characteristics make them more vulnerable. Its functionalities are characteristics of an *alarm system*. This system, also developed as an inter-organizational one, has not got ability to perform autonomous and collaborative corrective control actions.

Cauvin et al (2009) proposed an approach to minimize the impact of disruptive events on the whole intra-organizational information system. It is based on an analysis of disruptive events, the characterization of the error recovery process and a cooperative repair method for distributed industrial systems. The aim of this work is to assist human decision-makers in the design of the recovery process, proposing them solutions for the final decision. These functionalities are characteristics of a *decision support system*. The system is unable to perform autonomous corrective actions, and it was not designed to work into an inter-organizational context.

Guo and Zhang (2009) presented a Multi-agent based intra-organizational information system that can schedule manufacturing processes dynamically and flexibly responding to disruptive event generated by market demands. Its functionalities are characteristics of a *decision support system*. It is not provided with mechanisms to perform autonomous corrective actions to mitigate the effects of disruptive events, neither its architecture is adaptable to support an inter-organizational work.

In brief, Tracking and Tracing Systems are the status quo in most enterprises. Academic proposals as the ones discussed above are an evolution of these information systems, but the next step is a SCEM system able to perform autonomous implementation of solutions when a disruptive event occurs during operational plan execution. SCEM systems should be based on proactive and systematic methods of prediction and reaction to situations that are very different from the typical reports of exceptions generated by enterprise resources planning systems. The work presented in this paper is intended to contribute in this direction.

3. MULTI-AGENT BASED SCEM SYSTEM

Two main concepts used in this work are *plan* and *solution*. They are defined as follow:

Plan is a set of orders, each order representing a supply process (transformation or transference) that allocates materials to a place and stating the required resources, the time period during which each resource is required and its required capacity.

Solution is a set of control actions that uses the plan slacks to mitigate the effects of a disruptive event. When a solution cannot be automatically generated an exception occurs, which requires a re-planning task.

3.1.Model of main components of a SCEM system

To mitigate the effect of disruptive events in the plan that is being executed, a distribution of the variation among SC members, using the plan´s slack in a collaborative way, is proposed.

This proposal results in a distributed SCEM system working in an integrated way with the Planning Systems and the Execution Systems of each member of a SC. Fig. 1 graphically represents the main components of the resulting system for two SC members (A and B). A network of SCEM systems reflecting different organizations in a SC will be generated.

Planning Systems of SC members through a collaborative planning process agree on a supply plan. Based on that plan, the Planning System of each SC member generates an enterprise plan and communicates it to both the Execution System and the SCEM system. During the plan execution, SCEM system monitors changes from the Execution System and analyzes them to detect disruptive events. When it detects a disruptive event, search for an internal solution. If an internal solution can not be found the SCEM system may resort to the SCEM System of another organization for collaborative participation by executing a Collaborative Process. If a solution is found, the SCEM system sends it to the Execution System. Else, it notifies an exception to the Planning System for re-planning decisions.

Figure 1: Model of Main Components of a SCM system

3.2. A conceptual model of a SCEM system

The SCEM system of an enterprise has to support a complex control problem in which three types of variables can be identified: *observed variable* (observed during the plan execution with the purpose of detecting significant changes that may produce a disruptive event), *controlled or state variable* (defines a control point; it has a plan with slacks defined by the Planning System) and *decision variable* (independent variable whose value can be adjusted to find a solution with the purpose of bringing the system back to the specified objectives). To address complexity, the problem is decomposed into a set of simpler interwoven sub-problems.

Figure 2: Conceptual Model of the SCEM System

Intra-organizational and inter-organizational interaction types of the SCEM system are identified in the Model of Main Components (Fig. 1).

Intra-organizational interactions take place due to the link of the SCEM system with both the Planning System and the Execution System. To carry out this function, entities PAGE (interface with the planning system) and EVA (interface with the execution system) are specified in the Conceptual Model (Fig. 2). PAGE receives, from the Planning System, a Plan to be monitored and informs exceptions. EVA receives from the Execution System a Change and informs solution that the SCEM system has automatically generated for the Execution System to update its execution plan.

Inter-organizational interactions take place when the SCEM system of a SC member calls for collaboration to search for a solution. To carry out this function, an IOA (inter-organizational interface) entity is specified.

The changes observed in the order can be caused by changes in the resources or materials needed for its execution, that is, it is possible to define a causal relationship among disruptive events and material or resource changes producing them. Then, a control point is defined for each resource or material; that is, a sub-system in which control, state, and monitored variables are specified. These control points are connected among one another through supply processes. Then, *the Conceptual Model of the SCEM system is designed as a net of control points defined on resources or materials connected through supply processes*.

In the Conceptual Model (Fig. 2), a control point on a resource is called RKU (Resource Keeping Unit). A material inventory is a resource with particular characteristics, and thus it is modeled as a specialized control point called MKU (Material Keeping Unit). A supply process is called SP (Supply Process), and it is defined from an order that relates several control points (required resources).

The interface PAGE receives the Plan to be monitored and creates an RKU or MKU for each involved resource and relates them through the corresponding SP.

RKU has a requirement plan called Usage Agenda, in which all its assigned orders are detailed. This Usage Agenda can be represented by a 6-upla [id-order, start-date, duration, type, quantity, SP-name], where types (start, middle, end) are used to indicate when change occurs during the period. Based on its Usage Agenda, a RKU can generate its load profil*e* called *states plan* that can be interpreted as a function of its availability state. It indicates the required capacity of the resource and for how long it will be required.

MKU is represented by a 3-tuple [m, p, l], material (m), packing (p) and location (l), which allow its univocal identification. Each MKU has a requirement plan called Input/Output List where all orders of inputs and outputs of the material it represents are detailed. Based on its Input/Output List, a MKU can generate its states plan, which is a list of 2-tuples [inventory, date] that indicates the function of its availability state (inventory). In this way, *the Conceptual Model of a SCEM system is defined as a net of MKUs connected among them through SPs which uses RKSs to execute their tasks (transformation or transference).*

It is necessary to highlight that control points, RKUs or MKUs, do not directly interact among them but through the corresponding SPs, which they know by means of their Usage Agenda or Input/Output List, respectively.

Because of SP represents a transition among MKUs, three types of basic transitions can be defined from the basic attributes of MKUs represented in the 3-tuple [m, p, l]: material change (Δm) as a result of a chemical or physical operation; package change (Δp) as a result of a packing operation; and location change (Δl) as a result of a transfer operation. These basic transitions can be combined to obtain seven types of transformations: material change = Δm; package change = Δp; location change = Δl; material and location change = $\Delta m \cup \Delta l$; material and package change = $\Delta m \cup \Delta p$; package and location change = $\Delta p \cup \Delta l$; material, package and location change = $\Delta m \cup \Delta p \cup \Delta l$.

SP represents a materials balance process defining how MKUs are related and which RKUs are required through its bill of requirements. Each SP has an Activity Plan that can be represented by a 3-upla [start-date, duration, resource-list], where resource-list can be represented by a 5-upla [id-resource, start-date, duration, quantity, mode], where mode (consumption, production, use) is used to indicate if the resource is consumed, produced or used.

Each RKU is defined as a control problem where: the state variable is its availability state defined by its states plan; the monitored variable is its states plan; and the control variables are time in its Usage Agenda, which can be used to extend, bring forward and/or delay use requirements, and the resource capacity within the limits defined by the plan slack. The function of an RKU is managing the disruptive events that could alter the execution of the plan assigned to the resource it represents. For that reason, it must monitor the availability of this resource. To carry out this task, each RKU is registered in EVA, which notifies any significant Change received from the Execution System.

The disruptive events that can affect a RKU are classified into internal and external events. An internal event is detected by an availability change of the resource a RKU represents. When a change occurs, the RKU generates the new states plan (load profile) of the resource and analyzes it to detect whether a disruptive event has occurred. In this case, the RKU tries to find a Solution by using its control variables (time and resource capacity). If it finds a solution, which does not affect the orders associated to the resource, it implements the solution. The Implemented Solution is sent to the Execution System through EVA. When the disruptive event affects at least one resource-associated order, the RKU must search for a solution by resorting to a collective behavior. If a solution is not found, the RKU is responsible for notifying the produced exception to the Planning System through PAGE.

An external event for a RKU has its origin in a SP that requires its use. This disruptive event on the SP is a consequence of a disruptive event that affected some other SP-associated resource and that could not be solved with the slacks of the Usage Agenda of the RKU that represents it. In other words, they are disruptive events caused by the collective behavior in the search for a solution.

As it is a specialization of the RKU, an MKU can also receive two types of disruptive events. Internal events are related to a change of its inventory; for example, damaged or expired materials and inventory level updating. External events refer to changes both in time and quantity of some order of its Input/Output List. Both types of changes modify its states plan, which is the function of its availability state (inventory). Once a change of this kind occurs, the MKU must proceed as previously described for RKU.

When collective behavior in the search for a solution to a disruptive event requires the participation of another member of the SC, an inter-organizational interaction is implied and carried out through, IOA. For executing this task, IOA has a supplying orders plan agreed by the other member of the supply chain.

3.3.The multi-agent architecture for SCEM systems

Based on the Conceptual Model of the SCEM system described above, an agent-based architecture for the SCEM system was developed. Software agent technology was chosen because it is an innovative solution for distributed and autonomous systems, where the overall behavior emerges from interactions of its components (Pathak et al, 2000; vanDam et al, 2009).

The architecture proposed for the SCEM system for two SC members (A and B) (Fig. 3), is composed of two main types of agents, RKU Agent and SP Agent; and three service agents, EVA (EVent Agent), PAGE (Planning AGEnt) and IOA (Inter Organizational Agent). The agents have been designed following the BDI model (Rao and Georgeff, 1995). In this architecture, the functions of the MKU entity in the Conceptual Model (Section 3.2), modeling a material-inventory resource, are assigned to the most general RKU Agent.

Figure 3: Agent-based Architecture of the SCEM System: Components and Interactions

Using AUML (Agent Unified Modeling Language) the main roles, knowledge, behavior and perception components of the agents were specified (Bauer et al, 2001). As example, Fig. 4 presents an AUML Class diagram of the RKU Agent. In this diagram, a states machine describes the process the agent has to perform to carry out the previously described function.

Figure 4: AUML Class Diagram of RKU Agent

The RKU Agent is conceived as a manager of the resource it represents. Its main role is EventManager, to carry out this role also plays the roles InitiatorCoordination (when receiving an internal event) and ParticipantCoordination (when receiving an external event). To fulfill its roles needs to know its monitored, state and control variables, the usage agenda, the states plan, the initialization parameters, internal and external events, and the proposed and implemented solutions.

The RKU Agent is created by the agent PAGE. In the Initiating state, from the agent PAGE receives the information about its usage agenda, variables and parameters. The RKU Agent should send a message for registration to agent EVA informing its monitored variables. Afterward, the RKU Agent is in the Waiting Message state, while, simultaneously, AnalyzeAgenda cyclical behavior is executed to determine if there is an inactivity time to justify that the agent goes to sleep; in this case, the agent EVA is notified of when can be awakened. Thus, one of the messages received by the RKU Agent is wake up to be active again, and the other messages that it can receive are related to changes. Depending on the change type, the RKU Agent can evolve to Managing internal event or Managing external event.

When a change occurs, the RKU Agent registers the change in its states plan and analyzes whether the change leads to an internal event. If so, the RKU Agent goes to the Analyze internal event state, which is associated with the InitiatorCoordination role. If a solution is found notifies the agent EVA; otherwise, it must notify the exception to the agent PAGE. When receives an external event, goes to the Analyze external event state, which is associated with the ParticipantCoordination role. It analyzes the change proposals and decides to accept the participation or reject them.

3.4. The coordination process

The SCEM system has been defined in the Conceptual Model (Section 3.2) as a network of MKUs linked by SPs that uses RKUs for its realization. In the agent-based architecture of the SCEM system (Section 3.3), RKUs (and their specialization MKUs) are agents that can become initiators and participants in a process of finding and implementing a solution to a disruptive event. The process of coordination is defined as a mechanism for re-allocation of resources using their slack in the plan (Bartschi, 1996). This coordination process is based on a protocol used for exchange information. The Contract Net Protocol is the most widely used technique for coordinating the allocation of resources and tasks among a group of actors (Smith, 1980; Oprea, 2003). Among the proposals based on the Contract Net Protocol, the

Mediated Contract Net facilitates the dynamic creation of clusters of agents and provides collaborative transactions (Leitao, 2004).

To simplify the representation, each RKU Agent of the proposed SCEM system architecture only know the SP Agents it is related to (this information is contained in its Usage Agenda). With this organization, initiator and participant roles are carried out by RKU agents, while the mediator role is played by SP Agents when required.

This characteristic made it necessary to define an extension to Contract Net protocol based on the concept of Mediated Contract Net protocol, which has been called Double Contract Net Protocol (Fig. 5). This new protocol was designed to support an interaction through a Contract Net protocol between an RKU Agent as initiator and an SP Agent as mediator. The SP Agent, then, by using the information contained in its Activity Plan, initiates more Contract Net protocols to interact with other RKU Agents as participants. In this protocol, the RKU Agent that starts the process is called *coordinator*, and those RKU Agents that are contacted by the SP Agent are called *responders*. The SP Agent must perform both roles and is named *mediator*. The negotiation process can be seen in Fig, 5. Negotiation finishes when a solution is found or the deadline is reached. When a solution is reached, the coordinator agent notifies the other affected agents.

Figure 5: The Double Contract Net Protocol

3.6 Negotiation model

The negotiation between agents is carried out following the Double Contract Net Protocol. As negotiation moves forward, each RKU Agent completes the business document associated with the exchanged messages, and SP Agent performs quantity conversions and unifies business documents coming from different RKU Agents by using the bill of resources.

A proposal is composed of a list of alternatives, and each alternative is made up of a list of order-changes that represents the different changes that must be consolidated on a specific order. Then, it can be represented by a 2-tuple [alternative, order-change]. An order-change is represented by a 7-tuple [originator, receiver, quantity, time, utilitiesOriginator, utilitiesReceiver], where: originator is the name of order change originator; receiver is the name of order change receptor; quantity is the list of quantity variations (ΔQi) of order change; time is the list of time variations (Δtj) of order change; utilitiesOriginator is a list of Utility(ΔQi, Δtj), each them representing the matrix of utilities corresponding to quantity and time variations of the originator; and utilitiesReceiver is a list of Utility(ΔQi, Δtj), each them representing the matrix of utilities corresponding to quantity and time variations of the receiver. Each order-change contains ranks in which order quantity or time can vary. The lists of quantity and time variations model these ranks. Each Utility(ΔQi, Δtj) contains a numerical value which expresses the value of the utility function of originator/receiver for the corresponding quantity and time variation. That value is completed by each agent and is determined by its utility function.

The fact of defining list of values [ΔQ1, ΔQ2, ..., ΔQn] and [Δt1, Δt2, ..., Δtm] does not imply that only those values can be taken into account during negotiation. Actually, any value of quantity variation within the [ΔQ1; ΔQn] rank and any value of time variation within the [Δt1; Δtm] rank can be agreed on by the end of negotiation. To work this way, the Utility(ΔQi, Δtj) function is assumed to vary linearly for any value of ΔQi within the [ΔQi; ΔQi+1] subrank and for any value of Δtj within the [Δtj; Δtj+1] subrank. Thus, the Utility(ΔQi, Δtj) value for each point contained in each region can be calculated by linear interpolation.

4. SCEM SYSTEM VALIDATION

4.1 Methodology

As a preliminary validation of the SCEM model a performance analysis of the SCEM system through a case study has been performed. A case study offers the opportunity to study a phenomenon in its own natural setting where complex links and underlying meanings can be explored, whilst also enabling the researcher to study whole SC (Miles and Huberman, 1994; Yin, 1994). It is also appropriate where existing knowledge is limited because it generates in-depth contextual information which may result in a superior level of understanding (Oke and Gopalakrishnan, 2009). To this aim, a common SC case study was selected.

4.2 The case study

The case study, based on real enterprises, consists of a product distribution process involving a SC integrated by a supplier, a wholesaler with a central warehouse and a branch warehouse, and two retailers. The products distributed are: p1, supplied in two packages, p11 in package of 0.20 Tons and p12 in package of 0.025 Tons; and p2 supplied just in one package of 0.05 Tons.

The maximum capability of the warehouses varies from 80 to 200 Tons for each product into each node. Each member is in charge of storing these products and there is a transport process (location change (Δl)) among them. The SC has 20 semi-trailer trucks with capacity varying from 20 to 35 Tons that can transport approximately 3000 Tons a week between the nodes. An order shipped from a wholesaler's warehouse arrives to the retailer's warehouse in the same period (day). The attribute of the Usage Agenda Type = start was used; this indicates the states change occurs at beginning of period. In this SC, disruptive events may have their origin in the inventories (expiration date, breakdown, etc.) and resources (unavailable). The planning horizon is one week. The SC members collaboratively elaborate their supply plans, and based on that plan, the Planning System of each SC member generates an enterprise plan. The amount of all planned dispatch orders between nodes of the SC varies proximately between 80 and 130 orders per week.

4.3 Data collection

Data were collected through visits to the SC member's sites. The operational plans and the bill of resources were obtained from the Enterprise Planning System (ERP) data base. But other data set needed to perform this validation were disruptive events commonly present in this type of SC. The enterprises have not got an electronic record of them therefore they had to be obtained through interviews with the responsible of the logistics area of each SC member. They reported short-term frequent interruptions or delays that may occur daily in various parts of the SC. Based on the Conceptual Model of a SCEM system defined in Section 3.2, which was designed as a net of control points defined on resources or materials, these interruptions or delays were classified into *resource unavailability* and *inventory variation*.

The most frequent reported causes of unavailability of resources are produced by: i) Road-blocks generated by striking workers protests, which may be announced or sudden. These are often of short duration, usually from 2 to 4 hours. Only exceptionally may be long lasting as it did in 2008, where an agricultural protest blocked the routes of Argentina for 3 months. ii) Road closedowns due to fog, which are short time disruptive events usually from 2 to 3 hours that occur in autumn and winter. iii) Truck failure, which may occur even when preventive maintenance on trucks is performed. Repair or replacement of the truck can take 1 to 5 hours. iv) Lack of an operator by sudden illness, which may require several hours or even a day to get a replacement.

The most frequent reported causes of unexpected inventory variations are produced by: i) Inaccurate forecasts of customer demand that may imply unexpected variations of the inventory into retailers. ii) Rupture of containers during the transport or when they are manipulated into the warehouse. A particular case occurs when 0.20 Tons pack suffers damage; the product p1 that will not spill is transferred to packs of 0.025 Tons. That is, the product p11 becomes to p12. Then, inventory adjustments reducing the amount of product p11 and increasing the amount of product p12 are performed. iii) During the summer, quality problems can also occur in this SC, due to a chemical reaction taking place during travel and storage may alter the product specification. It can produce significant change of the inventory requiring a collaborative negotiation among trading partner with the aim to prevent an exception occurs.

4.4 An example of the SCEM system instantiation

The Multi-Agent SCEM prototype system has been implemented using JADE (Bellifemine et al, 1999). By instantiating the product distribution process in the system, an executable agent-based model of this supply process was generated. The model has been used to evaluate the behavior of the product distribution process to different disruptive events. For example, the Wholesaler operational plan is represented by a 7-upla [id-order, date, origin, destination, product, quantity, resource] where each record of the plan corresponds to an order. When the agent PAGE in the Wholesaler SCEM system receives this plan, it generates the different agents of the SCEM system. For example, for a particular record of this plan, which correspond to order number 35 requiring shipping 20 Tons of product p12 from the central warehouse to the branch warehouse using the truck2, the following four agents were generated: mku-p12-c representing the product p12 in the central warehouse, mku-p12-b representing the product p12 in the branch warehouse, rku-t2-c representing resource truck2 in the central warehouse and sp-35-c representing the transference supply process of order number 35 from the central warehouse.

Once instantiated, the SCEM system started its monitoring function of the plan in execution. For example, a summer day with high temperature, 5 Tons of product p12 were returned to the supplier due to a quality problem, the agent EVA received a message informing that the product p12 initial inventory in the central warehouse was 65 Tons instead of the planned 70

Tons. EVA sent a message to agent mku-p12-c informing the change. This agent generated the states plan after the change and compared it with the previous states plan. Fig. 6 shows a graphic representation of both states plans. Comparing them, the agent could anticipate that during the fifth day the inventory will be less that the minimum value. In this way it found out that a disruptive event occurred. Because of the agent mku-p12-c assumed the role of coordinator in the process of searching for a solution.

Figure 6: mku-p12-c State Plan before and after the initial inventory change

To find a solution, the process can evolve in three stages: (1) seeking a solution by working only with the orders of the period where the disruption has occurred; (2) considering the orders of the periods immediately before and after the period in which the disruption has occurred; (3) considering all orders from current time. At each stage, the process can select one of three steps: (a) modifying only one parameter of one order (quantity or time); (b) changing both parameters of one order (quantity and time); (c) combining variations in more than one order.

Following this process for seeking a solution (stage 1), the agent mku-p12-c selected the order number 27 scheduled to be executed on the fourth day requiring shipping 30 Tons of product p12 from the central warehouse to the retailer1 using the truck3. Using the Double Contract Net Protocol, the agent mku-p12-c agreed on with the agent mku-p12-m1 representing the product 12 into the retailer1 and the agent rku-t3-c representing the truck3 to modify the order quantity from 30 to 25 Tons (step a). The agent ps-27-c representing the transference supply process of order number 27 from the central warehouse has been the mediator. Fig. 7 shows a graphic representation of both states plans disrupted and repaired for the agent mku-p12-c, and of both states plans previous and subsequent for the agent mku-p12-m1 that participated of the negotiation. As can be see in Fig.7 (above), in the repaired states plan of the agent mku-p12-c the inventory during the fifth day will be equal to the minimum value, and in the states plan of the agent mku-p12-m1 the inventory during this period was reduced 5 Tons, reaching the minimum value. The states plan of the agent rku-t3-c has not been affected. The solution was sent to the Execution System through EVA for its implementation.

Figure 7: Initiator (mku-p12-c) (above) and participant (mku-p12-m1) states plans

In short, once detected the change, the agent-based system was able to analyze its impact by projecting it along the time horizon of the operational plan (a week). In this way the system detected a disruptive event that would produce a stock out on the fifth day. Based on this evidence, the affected agents carried out a collaborative process that allowing negotiates a solution to mitigate the disruptive event effects. The solution consisting into modifying from 30 to 25 Tons the quantity of the order number 27 that should be dispatched on the fourth day. The modified operational plan was sent by the SCEM system to the Execution System for its implementation. All the process has been carried out without the human participation.

4.5 Results analysis

The example is based on a scenario defined by a supply process with a simple bill of resources, where a solution may be easy to obtain for a decision maker. Despite its simplicity, it has enough richness to proof the concept that the proposed mechanisms are able to obtain and implement the solution automatically. This accomplishes the main objective of this work, which is to avoid the need for human intervention when the problem can be solved automatically.

The system architecture and its mechanisms of analysis are independent of both the type of the supply process and the dimensionality of the bill of resources associated to it. Therefore, any of the seven types of supply process that where defined in Section 3.2, with a bill of resources involving a finite amount (hundreds) of resources can be instantiated (modeled) in the system. The dimension affects the efficiency (time of response) of the system but not it efficacy.

A preliminary analysis of the system efficacy was performed. The solution proposed by the SCEM system for each disruptive event was evaluated for the responsible of the logistics area of each SC member. Only the 64% of these were considered as satisfactory. The main factor influencing this result is the estimation of utilities used by the negotiation model (Section 3.6).

A systematic analysis of the system efficiency was not done yet. But, SP agents representing transformation processes with a list of up to fifty elements (resources) conducted the negotiation processes in a few seconds.

5. Conclusion and Future Work

Risks can affect a SC on different way. A machine failure may have a low impact on the enterprise with enough redundant capacity, whilst a natural disaster as the last Chilean earthquake may have a high impact on a SC. The big challenge for managers is to mitigate risk by intelligently positioning and sizing SC reserves without decreasing profits (Chopra and Sodhi, 2004; Craighead et al, 2007; Trkman and McCormack, 2009; Tang, 2006; Tuncel and Gülgün, 2010). Risks of high impact are the focus of strategic decision processes while risks of low impact are the focus of operational decisions.

Risks of low impact produced by frequent short term disruptions or delays occur daily on several points of the SC and affect not only the organization where they are produced, but also propagate throughout the SC. To mitigate this type of risks, planning systems generate robust and flexible operational plans by including slack (material, resource capacity and time reserves). This allows adapting an operational plan to the conditions that occur during its execution, provided that the interruption or delay does not exceed the slack.

The approach presented in this work is focused on, and therefore is limited to, risks of low impact produced by frequent short term disruptions or delays. The presented agent-based approach for the SCEM problem offers a solution for allowing automating the process of mitigating the effect of changes in the currently running operational plan through a distribution of changes among SC members by using the plans slack in a collaborative way. In this way, the SC agility and ability to respond to disruptive events are improved.

To define the reserves (slack) planning systems implement appropriate mitigation strategies (Chopra and Sodhi, 2004; Schoenthaler and Alvarenga, 2003). The design of these strategies is out of the scope of this paper.

The mechanisms provided for the system to perform autonomous corrective control actions were included in response to the requirement of ability to exert corrective control actions identified as an area barely explored (Zimmermann, 2006).

An innovative feature of this approach is its focus on resources, which are the ones affected by disruptive events in a direct way, and not on the orders, which are the ones affected by the effects triggered by such events. Based on this approach, the conceptual model of the SCEM system was designed as a net of control points defined on resources connected through supply processes order, which can address the complexity of the involved control problem, decomposing it into a set of simpler interwoven sub-problems defining a control point in each source of primitive events.

A novel aspect is the distributed collaborative inter-organizational architecture of the proposed SCEM System. This addresses the need for an approach to disruptive events management taking into account the distributed nature of SC and respecting members' autonomy (Cauvin et al, 2009).

Another novel aspect proposed in this research work is the Double Contract Net Protocol. This consists of a protocol a coordinator agent starts by executing a Contract Net protocol with a mediator agent, which in turn starts many Contract Net protocols with so many responder agents as necessary. This protocol allows a set of RKU Agents to interact with them through a SP Agent as a mediator.

The solutions generated by the SCEM systems implemented using JADE through validation proof were evaluated as satisfactory by the domain experts in only 64% of the cases. Therefore, future work should contemplate the development of new algorithms to improve the negotiation model, specifically to improve the generation of the solution space and the ability to select best solutions. Another aspect future work should contemplate is to improve corrective control abilities of SCEM systems adding new algorithms based on complex event processing (Luckhem 2010). Formal modeling languages as colored Petri nets can be used for examine event causality and forecast subsequent events (Liu et al 2007). Finally, techniques for error recovery can be also used to develop the algorithms (Wyns, 1999; Ribeiro et al, 2009; Shen and Norrie, 1998; Monostori et al, 1998; Bussmann et al, 2004).

References

1. Adhitya, A., Srinivasan, R. and Karimi I.A., 2007. A model-based rescheduling framework for managing abnormal supply chain events. Computers and Chemical Engineering, 31(5), 496-518.
2. Bartschi, M., 1996. A genetic algorithm for resource-constrained scheduling. PhD Thesis. MIT.
3. Bauer B., Müller J. and Odell J., 2001. Agent UML: A formalism for specifying multiagent software system. Agent-Oriented Software Engineering LNCS, 1957/2001, 109-120.
4. Bellifemine F., Poggi, A. and Rimassa, G., 1999. JADE – A FIPA-compliant agent framework. CSELT Internal Technical Report. Part of this report has been also published in Proceedings of PAAM'99, London, April, 97-108.
5. Bodendorf, F., & Zimmermann, R., 2005. Proactive supply-chain event management with agent technology. International Journal Electronic Commerce, 9(4), 58-89.
6. Bui, T., Hempsch, C., Sebastian, H. and Bosse, T., 2009. A multi-agent simulation framework for automated negotiation in order promising. In Proceedings of the Fifteenth Americas Conference on Information Systems. August 6-9, San Francisco, California, USA.
7. Bussmann, S.; Jennings, N. R. and Wooldridge, M., 2004. Multiagent systems for manufacturing control: A design methodology. T. Ishida, N.R. Jennings, K. Sycara (Series Eds.). Springer Series on Agent Technology. Springer-Verlag, Berlin Heidelberg, Germany.

8. Cauvin, A.C.A., Ferrarini, A.F.A. and Tranvouez, E.T.E., 2009. Disruption management in distributed enterprises: A multi-agent modelling and simulation of cooperative recovery behaviours. In International Journal of Production Economics, 122(1), 429-439.
9. Chopra, S. and Meindel, P., 2001. Supply chain management: Strategy, planning, and operations, upper saddle river. Prentice-Hall, New Jork, USA.
10. Chopra, S. and Sodhi,M., 2004. Managing risks to avoid supply-chain breakdown. MITSloan Management Review. 46(1), 52-62.
11. Christopher, M., 2005. Logistics and supply chain management. Creating value-adding networks. Prentice Hall, New Jork, USA.
12. Craighead, C., Blackhurst, J. Rungtusanatham, M. J. and Handfield, R. 2007. The severity of supply chain disruptions: Design characteristics and mitigation capabilities. Decision Sciences, 38(1), 131-156.
13. Guo, Q. and Zhang, M. 2009. A novel approach for multi-agent-based intelligent manufacturing system. In Information Sciences, 179(18), 3079-3090.
14. Hoffmann,O., Deschner, D., Reinheimer, S. and Bodendorf, F., 1999. Agent-supported information retrieval in the logistic chain. In proceeding of the 32nd Hawaii International Conference on System Sciences, Maui, USA.
15. Kärkkäinen, M., Främling, K. And Ala-Ri, M., 2003. Integrating material and information flows using a distributed peer-to-peer information system. In Collaborative System for Production Management, 305-319, Kuwer Academic Publishers, Boston, USA.
16. Knickle, K. and Kemmeler, J., 2002. Supply chain event management in the field success with visibility, AMR Research, Boston, USA.
17. Leitao, P., 2004. An agile and adaptive holonic architecture for manufacturing control. PhD Dissertation. Department of Electrotechnical Engineering Polytechnic Institute of Braganca, University of Porto, Portugal. Online: http://www.ipb.pt/pleitao
18. Liu, Q. and Min, H., 2008. A collaborative production planning model for multi-agent based supply chain. International Conference on Computer Science and Software Engineering. Wuhan, China.
19. Liu, R., Kumar, A., & van der Aalst, W., 2007. A formal modeling approach for supply chain event management. Decision Support Systems, 43(3), 761-778.
20. Masing, N., 2003. Supply chain event management as strategic perspective – market study: SCEM software performance in the european market. Master Thesis. Universitié du Québec en Outaouasis.
21. Meydanoglu, E., 2009. The rol of supply chain event management systems for supply chain risk management. In European and Mediterranean Conference on Information Systems 2009 (EMCIS 2009). July 13-14. Izmir, Turkey.
22. Miles, M.B., and Huberman, A.M., 1994. Qualitative data analisys. A souce books of new methods. Sage, Newbury Park, California, USA.
23. Monostori L, Szelke, E. and Kadar, B., 1998. Management of changes and disturbances in manufacturing systems. Annual Reviews in Control, 22, 85-97.
24. Oke, A., and Gopalakrishnan, M., 2009. Managing disruptions in supply chains: A case stugy of a retail supply chain. Intenational Journal of Production Economics, 118(1), 168-174.

25. Oprea, M., 2003. Coordination in an agent-based virtual enterprise. In Studies in Informatics and Control, 12(3), 215-225.
26. Pathak, S., Nordstrom G. and Kurokawa S., 2000, Modeling the supply chain: A multiagent approach. In Proceeding of IEEE International Conference on System, Man and Cybernetic, 3, 2051-2056.
27. Pereira, J., 2009. The new supply chain´s frontier: Information management. In International Journal of Information Management, 29(5), 372-379.
28. Rao, A. and Georgeff, M., 1995. BDI Agents: From theory to practice. Proceedings of the First International Conference on Multi-Agent System. June 12-14, San Francisco, USA.
29. Ribeiro, L., Barata, J. and Colombo, A., 2009. Supporting agile supply chains using a service-oriented shop floor. In Engineering Applications of Artificial Intelligence, 22(6), 950-960.
30. Schoenthaler, R and Alvarenga, C., 2003. A new take on supply chain event management. In Supply Chain Management Review. March/April, 29-35.
31. Shapiro, J., 2000. Modelling the supply chain. Duxbury Press, Pacific Grove, California, USA.
32. Shen, W. and Norrie, D.H., 1998. An agent-based approach for manufacturing enterprise integration and supply chain management. Proceedings of the 3rd International Conference on the Practical Applications of Agents and Multi-Agents Systems. London, UK.
33. Simchi-Levi, D., Kamanisky P., and Simchi-Levi E., 1999. Designing and managing the supply chain. Mc Graw Hill, USA.
34. Smith, R. G., 1980. The Contract Net Protocol: High-level communication and control in a distributed problem solver. IEEE Transactions on Computers, 29 (12), 1104-1113.
35. Speyerer, J. and Zeller, A., 2004. Managing supply networks: Symptom recognition and diagnostic analysis with web services. Proceedings of the 37th Hawaii International Conference on System Sciencies. Hawaii, USA.
36. Swaminathan, J., Smith, S and Sadeh, N., 1998. Modeling supply chain dynamics: A multiagent approach. Decision Sciences, 29(3), 607-632.
37. Szirbik, N.B., Wortmann, J.C., Hammer, D.K., Goosenaerts, J.B. M. and Aerts, A.T.M., 2000. Mediating negotations in a virtual enterprise via mobile agents. In Proceedings of the Academia/Industry Working Conference on Research Challenges (IEEE Computer Society Eds.), 237-242, Buffalo, New York, USA.
38. Tang, C., 2006. Perspectives in supply chain risk management. In International Journal of Production Economics, 103(2), 451-488.
39. Teuteberg F and Schreber, D., 2005. Mobile computing and auto-ID technologies in supply chain event management – An agent-based approach. In Proceedings of the Thirteenth European Conference on Information Systems, September 21-25, Regensburg, Germany.
40. Trkman, P. and McCormack, K., 2009. Supply chain risk in turbulent environments – A conceptual for managing supply chain network risk. In International Journal of Production Economics, 119(2), 247-258.
41. Tuncel, G. and Gülgün, A., 2010. Risk assessment and management for supply chain networks: A case study. In Computers in Industry, 61(3), 250-259.

42. Van Landeghem H. and Vanmaele, H., 2002. Robust planning: A new paradigm for demand chain planning. In Journal of Operations Management, 20(6), 769-783.
43. vanDam, K., Adhitya, A., Srinivasan, R. and Lukszo, Z., 2009. Critical evaluation of paradigms for modelling integrated supply chains. In Computers and Chemical Engineering. 33(10), 1711-1726.
44. Wang, L. and Lin, S., 2009. A multi-agent based agile manufacturing planning and control system. In Computers and Industrial Engineering, 57(2), 620-640.
45. Wyns, J., 1999. Reference architecture for holonic manufacturing systems. PhD-Thesis. Katholieke Universiteit Leuven. Belgium.
46. Yin, R. K., 1994. Case study research: Design and method. second ed. Sage, Thousand Oaks, California, USA.
47. Zimmermann, R., 2006. Agent-based supply network event management. Whitestein Series in Software Agent Techonologies, Eds: Marius Walliser, Stefan Brantschen, Monique Calisti y Thomas Hempfling. Birkhäuser Verlag, Switzerland.

>We present an autonomous collaborative distributed agent-based approach for SCEM problem.
>Autonomous control actions minimize the disruptive event effect using plan slack.
> SCEM System Model is a net of control points on resources connected through supply processes.
> The coordination process is based on Double Contract Net Protocol, it re-allocates resources using plan slacks.
>An application to a case study of SCEM System implemented with JADE is provided.

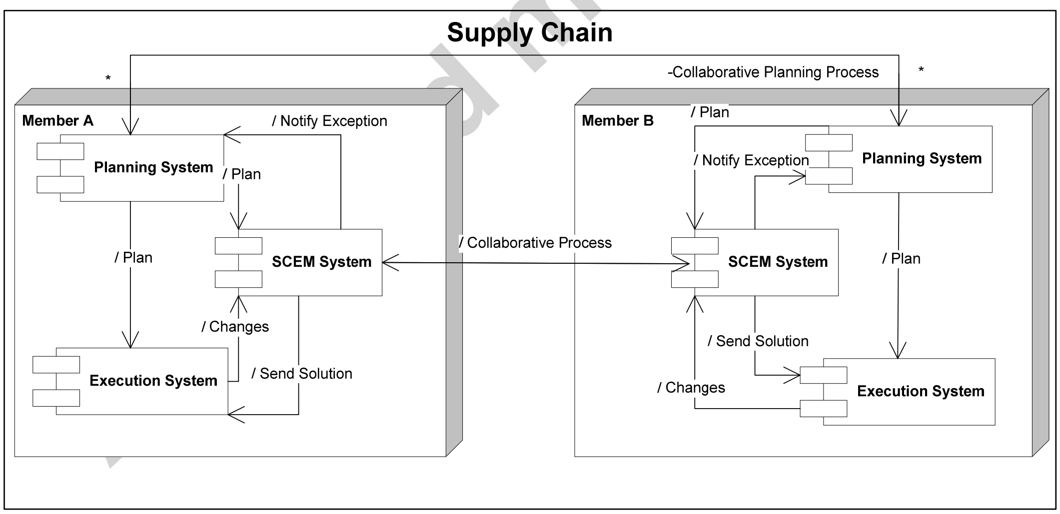


Figure 1: Model of Main Components of an SCM system

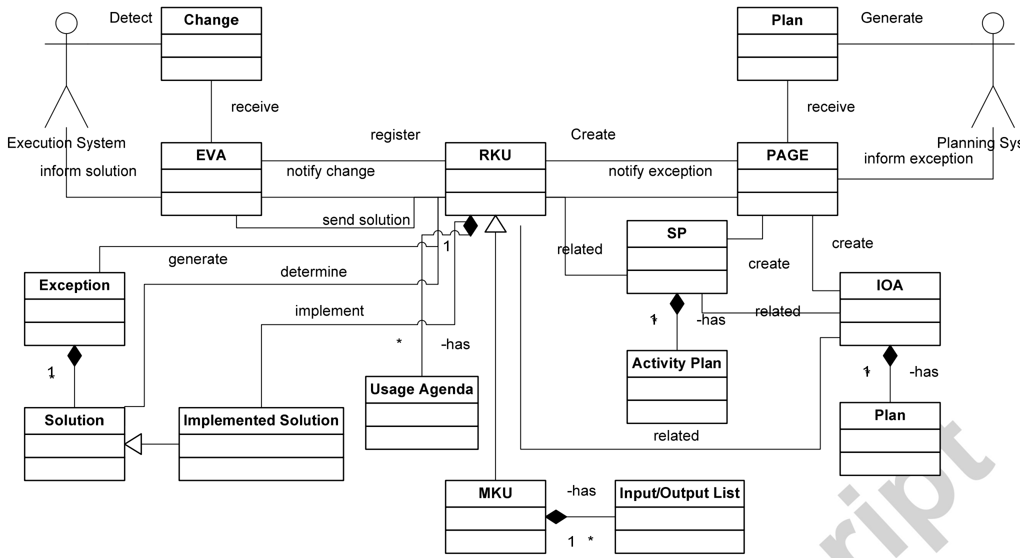


Figure 2: Conceptual Model of the SCEM System

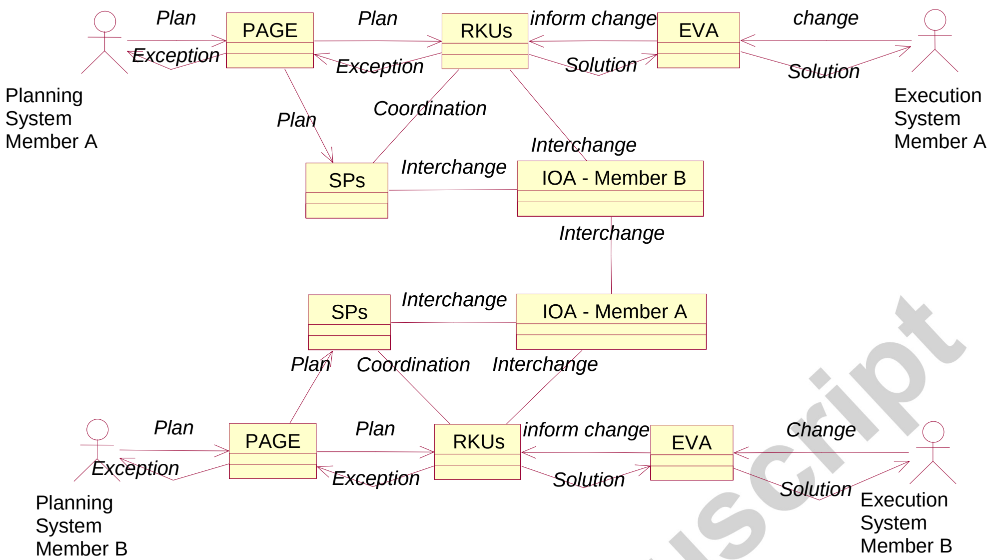
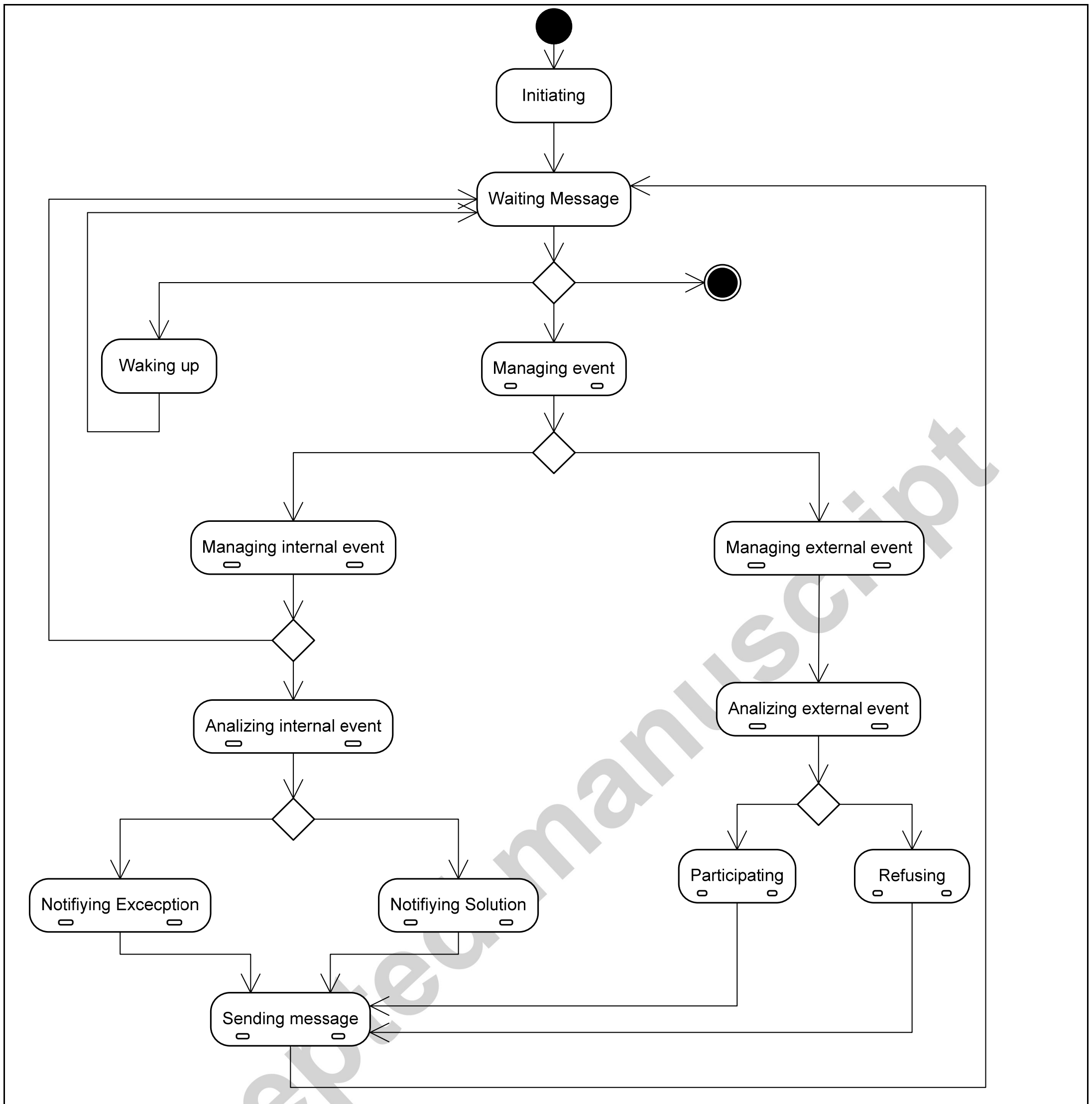


Figure 3: Agent-based Architecture of the SCEM System: Components and Interactions

<<agent>> RKU
Roles EventManager, InitiatorCoordination, ParticipantCoordination
State Machine



Knowledge

Internal Event, External Event, Proposed Solution, Implemented Solution, Usage Agenda, States Plan, initial parameter, monitored variables, state variables, control variables,

Behavior

Reactive [newmessage] WaitforMessages [messageprocessed]
Reactive [event-end] CloseAgent [deleteRKU]
Reactive [event-internal] ManageInternalEvent [solution|exception]
Reactive [event-external] ManageExternalEvent [participate| not participate]
Reactive [wakeup] WakeupRKU [rkuactive]
Cyclic [] AnalizeAgenda [continue | sleep]
Internal [event-internal] CoordinateSearchforSolution [solution|exception]
Internal [event-external] ParticipeExternalEvent [solutionsproposed]
Internal [sleep] NotifySleep [rkusleeping]
Internal [call] SendMessage [messageSent]

Perception

ACL-Message (Inform-instantiation| inform-eventinternal | cfp-eventexternal | inform-wakeup | inform-delete)

Figure 4: AUML Class Diagram of RKU Agent

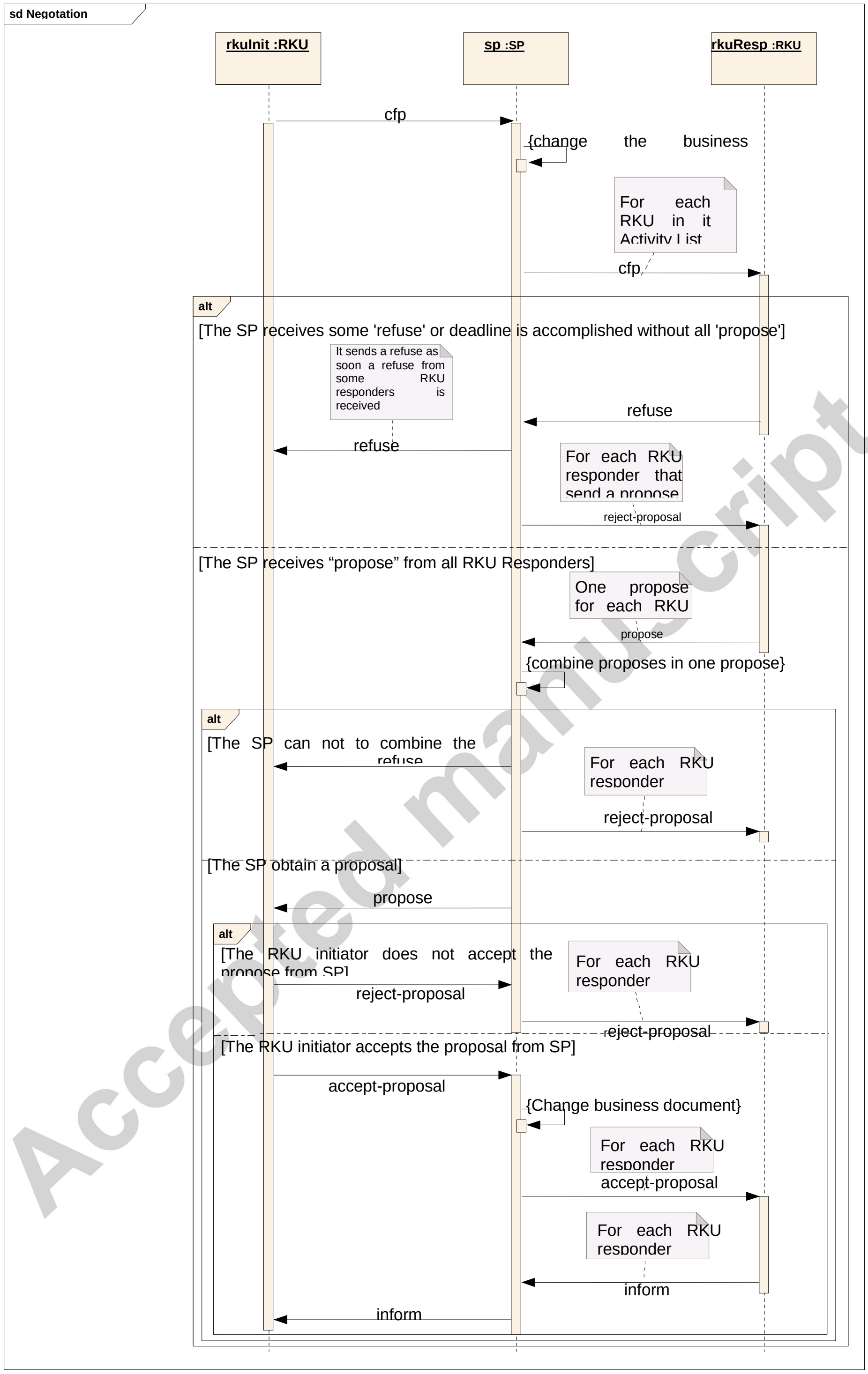


Figure 5: The Double Contract Net Protocol

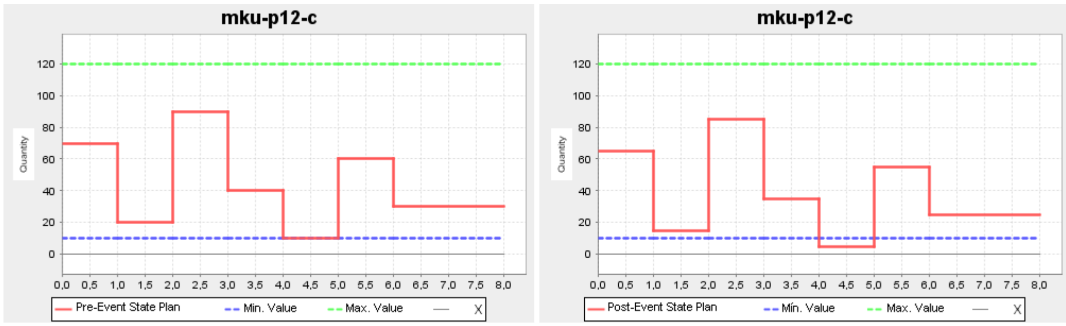


Figure 6: MKU-P12-C State Plan before and after the initial inventory change

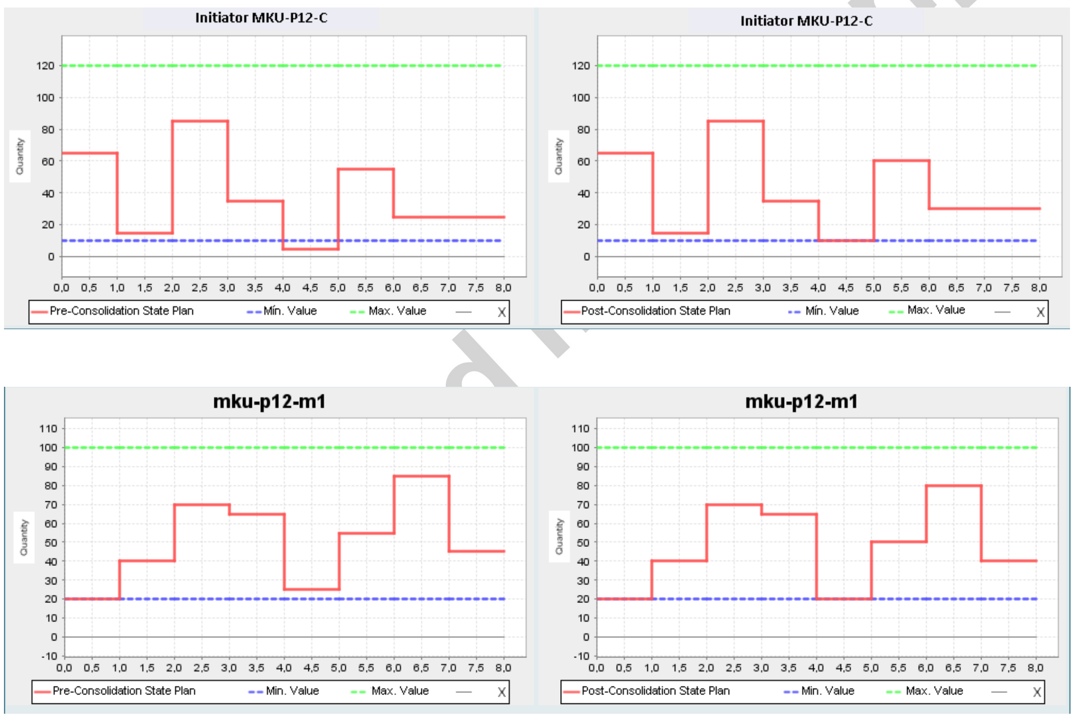


Figure 7: Initiator (mku-p12-c) (above) and participant (mku-p12-m1) states plans.