Service Oriented Approach for Autonomous Exception Management in Supply Chains

Armando Guarnaschelli, Omar Chiotti, Enrique Salomone

Service Oriented Approach for Autonomous Exception Management in Supply Chains

Armando Guarnaschelli[1], Omar Chiotti[1], Enrique Salomone[1]

INGAR-CONICET, Avellaneda 3657, S3002GJC
Santa Fe
{guarnaschelli, chiotti, salomone }@santafe-conicet.gov.ar

Abstract. Risk and uncertainty are inherent to Supply Chains; at the execution level unexpected events can disrupt the normal flow of supply processes creating a gap between planned operations and what is actually executed. These disruptions increment rescheduling frequency, generating reconfiguration costs and system's nervousness. This work proposes a web service based Business Process to support Autonomous Exception Management in Supply chains.

Keywords: Supply Chain, Exception, Rescheduling, Web Services

1 Introduction

A supply chain is an event driven system and requires the task of execution control. Either done manually or through an Execution Control System this task includes the monitoring of events during the execution of an scheduled set of supply process orders, detecting unexpected events, and alerting of exceptions caused or likely to be caused by some of them. An exception can be defined as a deviation in the execution that prevents the fulfillment of one or more of these supply process orders. Within a supply chain an exception not only affects its epicenter but also propagates throughout and many times amplifying its effects as it goes farther away.

In presence of an unexpected event, the execution system has to identify its source and communicate it to all interested parties within the supply chain. If the execution schedule is robust enough the variation caused by the event can be absorbed, and execution continues, if not the execution schedule becomes invalid and consequently this triggers a rescheduling task. Current Supply Chain Management Systems lack of systematic approaches to exception/disruption management. This is a deficiency that needs to be addressed to preserve a supply chain competitive in the future landscape. As stated in [1] future supply chains will be more adaptive (able to react quickly and correctly to changes) and unexpected events will be managed and contained on site making rescheduling activities less frequent.

There are substantial benefits in repairing execution schedules rather than triggering full rescheduling activities. The methods found in the literature do not explicitly exploit the slack provided to resources in their execution schedule. If these slacks were systematically exploited, the ability to repair a disrupted schedule would be increased. Moreover most methods to repair schedules we have found in the literature [2-4] do not consider the distributed nature of supply chains.

In this work we discuss a service oriented approach for supporting the business process of autonomous exception management in the context of supply chain execution control. We make an abstraction of the underlying monitoring and execution system and we adopt a service oriented approach to provide the functionality for handling exceptions and repairing disrupted supply processes, through standardized and self-contained descriptions that can be utilized by any sort of Supply Chain Execution System.

In particular, we present two essential components in the quest of building a service oriented solution to the mentioned problem. The first is an Exception Management Business Process, the second is a Feasibility Restoration Service.

These components are developed over the basis of a Reference Model for Supply Processes Exception Management, necessary to provide a general understanding of the problematic of execution control among supply chain business partners, previously described in [5]. This model has two main features: a) provides a self-contained description of any ongoing execution schedule of supply process orders with all the information required to assess its feasibility; b) it allows the automatic transformation into a constraint satisfaction program suitable for the autonomous search of local solutions for disrupted schedules.

These two features of the Reference model have been exploited to design an exception management business process, which autonomously supports the management of supply chain exceptions. In the context of this work the concept of exception management is addressed in a specific sense: automatically repairing a disrupted schedule with the advantages of: given an unexpected event, detecting future exceptions in advance across a whole supply chain, and everywhere possible avoiding the occurrence of exceptions making allowed local changes in execution schedules.

The business process relies on a service for feasibility restoration which uses instances of the reference model as the main business document to be exchanged with every partner's execution control process. The service encapsulates the functionality for translating this business document into a problem description suitable for automatic solution and return repaired schedules in the same form. We introduce the internal mechanism of this service which embeds a local repair algorithm for disrupted schedules. Finally an empirical validation of the proposed solution is given in form of a case study

2 Related Work

As Exception Management is a subject studied in many different areas we have identified two main areas closely related to our work, the generic area of complex software systems, and the specific area of Supply Chain Event Management (SCEM) Systems.

There is a parallelism between Exception Management in Supply Chains and exception handling in complex software systems such as Workflow Management Systems, Process Management Systems and Self-healing Systems. In the area of Workflow Management Systems the support for exception handling goes from the definition of exception handlers and methods for the specification of exceptional

behavior to classification and forecasting of exceptions in workflows [6-8]. In Process Management Systems the definition of exception handlers invoked under given conditions provide support for given types of exceptions [9-10]. And in the area of self-healing systems applied to service based applications and process management there are several approaches that diagnose faulty situations and to select and/or search for a recovery strategy [11-12]. While these approaches are useful frameworks for managing exceptions of general business processes, the nature of exceptions in the specific domain of supply chain processes can be exploited to build more powerful corrective actions to re-establish feasibility at the same time the exception is being handled.

In our work we intend to provide this additional feature by capturing, in a systematic way, the aspects of the supply process feasibility needed to automate a reparation mechanism.

Supply Chain Event Management is defined as the business process where significant events are timely recognized, reactive actions are quickly triggered, the material and information flows are adjusted and the key actors are immediately notified [13]. SCEM solutions according to [14] must implement the functions of: i) Monitor, providing on-going information about supply chain processes, workflows, and events, including the current status of inventories, orders, shipments, production, and supply. ii) Notification to relevant actors. iii) Simulation, supporting decision making by assessing what will happen if specific actions occur. iv) Control, letting a decision maker to introduce corrective actions, such as diverting a shipment or expediting an order. v) Measure, for assessing how well the supply chain performs.

Research in SCEM systems has mainly focused in addressing the monitoring, the capture and the communication of disruptive events. The ability to exert corrective control actions has been identified as an area barely explored [13, 15]. In our approach we allow the generation of a repaired execution plan that puts back on track the normal flows and also keeps alterations (actions needed to repair) to the original plan minimal (within planned slacks). The work of [16] gives a method to search for solutions to disruptions based on multi-agent negotiation. This approach do not consider the planned availability of resources versus resource utilization by orders), therefore the support of autonomic decisions is rather limited to give recommendations to a decision maker that will analyze them. Our work gives the basis for an autonomous exception management system automatically deriving repair actions (changes in a disrupted execution schedule) fully executable.

It is relevant to emphasize that the service oriented approach proposed in this work relies on the definition of an exception management business process that uses a Feasibility Restoration Service to generate the solution to the exception but does not require any specific architecture in the underlying SCEM system or execution control system in the supply chains that are affected by the exception.

3 An Exception Management Reference Model

3.1 Modeling Supply Processes.

Whenever an exception occurs it is important to track its origin and delimit its propagation, to be able to attenuate its effects or even eliminating them. To do this we describe an on going execution schedule and the possible sources and propagation paths of an exception as a net of Resources and Supply Process Orders (SPOs) linking them. We adopt this representation not only because it can show the origin and propagation of an exception, but also allows monitoring and controlling events and exceptions at their origin, communication of the exceptions to the proper receptor (whoever has control of the resources and SPO involved) through the Supply Process Orders and eventually the evaluation of exception's effects propagation and impact.

In (Fig.1) we provide a UML Class Diagram Representation of a general Supply Process, linked Supply Processes conform a net of Resources and SPOs.

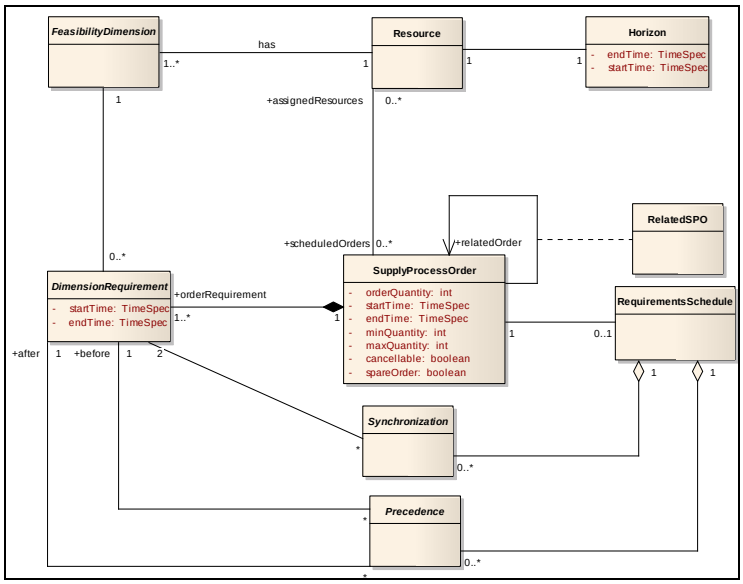

Fig. 1. Modeling Supply Processes

As stated in (Fig. 1) Class Diagram, a Supply Process is defined, through a SupplyProcessOrder (SPO), which is composed by a set of DimensionRequirements imposed to every FeasibilityDimension of the resources assigned for the execution of the SPO. When two SPOs belong to different business partners, their relation is captured by the association class RelatedSPO, which implies relationships between the two SPOs, such as same orderQuantity, same timing, etc.

Some SPOs, can be cancelled in favor of the feasibility of execution of other SPOs, and there can be special SPOs called spare, that are only executed in case of emergency, for example an SPO using a 3PL (third party logistics provider).

In a typical Supply Chain, an exception can cause different kind of losses, amongst them service level diminishment of the node causing the exception and in general for the whole supply chain. But the exception might not affect the totality of the on going execution schedule of the exception related nodes, but instead a set O of Supply Process Orders (SPO). Every SPO ε O is related to a set of resources required for its fulfillment belonging to any of the affected nodes. Whatever the exception, if the information required is on hand, it is possible to trace its origin to the unavailability of one of the related resources or to a change in one SPO.

3.2 Describing Resources by Feasibility Dimensions.

In order to assess a resource's availability, the feasibility of its schedule and to evaluate the effects of events and exceptions, it is necessary to describe resources. A possible attempt is to classify resources by types as in [17], but this has the drawback that resources in a supply chain can be quite diverse, a more generic and extensible characterization is needed. There is a widely known resource characterization developed by [18] and is intended to use for the specification of scheduling model in constraint programming. We choose to develop another characterization which is very general also, but specifically tailored for the purpose of repairing disrupted supply process across the supply chain. We don't use other resource models because we require more than a resource ontology, showing the semantics of resources, but also the syntactic properties with respect to resource utilization and we give them by describing resources by Feasibility Dimensions.

As our purpose is to model the availability of resources and the feasibility of its scheduled Supply Process Orders, we propose to characterize resources by introducing the concept of feasibility dimensions.

A Feasibility Dimension is a characteristic of a resource describing its capability to fulfill a requirement from a Supply Process Order. The availability of the resource is therefore conditioned by them and every requirement for the resource should be expressible in terms of its feasibility dimensions.

We define two types of feasibility dimensions, Capacitated Dimension and State Based Dimension.

A full description of this reference model can be found in [5]. It is relevant to emphasize that an instance of this Reference Model is a self-contained description of a feasibility problem in an execution schedule that can be automatically transformed into a Constraint Satisfaction Problem (CSP). This CSP, checks the feasibility of the execution schedule, and also gives the possibility of identifying slacks in order to device a repair mechanism with minimum modifications, as the one presented here.

4 Exception Management Business Process using a Web Service

The inter-organizational nature of Supply Chain Management requires coordination between different Supply Chain partners. Acknowledging this fact we rely the coordinated search for solutions given a disruption to a Specialized Compound Web Service called Feasibility Restoration Service (FRS). Autonomous Exception Management requires the existence of a Supply Chain Execution Control System, these Systems are usually specific and private for each Supply Chain partner. Therefore, a possible solution for this inter-organizational problem is using a third party FRS, that having the ability to understand Execution Schedules offers the functions of Exception Detection (evaluate the impact of an unexpected event) and automated repair of disrupted Supply Processes. This repair is done strictly using SPOs and resources explicitly scheduled slacks, and discovering implicit slacks, hidden in the interaction between SPOs and Resources.

The rationale of choosing a service oriented solution is that any effective business process to deal with exceptions in supply chains must be inter-organizational by nature. By adopting this type of architecture we can better address complex issues like the extent of information being shared among different parties, the control of the collaborative flows and the systems interoperability.

The FRS allows having a global view of the ramification of an exception across any number of business partners yet keeping the information of each partner isolated from the others.

4.1 Business Process Description.

The EMBP begins when the Supply Chain Execution Control System (SCEC) identifies an unexpected event. Assuming previous service contract between SCEC and FRS, a *Request for Evaluation* is sent to the FRS. This request includes a business document which is an instance of the Exception Management Reference Mode, that is a Net of Supply Processes. This net might be the full Supply Chain Schedule, or just a portion where the event emerged.

To evaluate the event, the minimal information shared must contain, the origin of the event together with the associated resources if it was a SPO, or the associated SPOs, if the origin was a resource.

Then the FRS evaluates the unexpected event using the *Evaluate Unexpected Event* Service, if the event does not affect the Net of Supply Process, the EMBP finishes, as the scheduled slacks absorbed the effects of the unexpected event. If not, and the schedule results to be infeasible a notification is sent to the SCEC, and this must decide whether to rely on the FRS to search for a solution to this exception, or generate an alarm, using the FRS as a feasibility evaluation service.

If the SCEC decides to rely on the FRS to search for a solution another stage of the EMBP takes place, in it the FRS tries to find a local feasible solution to the disrupted schedule using the RepairNet service. The effort to search for this solution is based on the incremental sharing of Schedule information from the SCEC. This information is shared until a solution is found, and then the SCEC implements the new feasible schedule or until a solution was not found and the net cannot longer be expanded, or

the SCEC is not willing to share more information. (Fig. 4) shows a BPMN [19] specification of the FRS. The aforementioned services Evaluate Unexpected Event and RepairNet are implemented using the CSP in [5]. In (Fig. 4) the relationship between the business document (an instance of the Exception Management Reference Model) and the FRS with these subservices is shown.

It is important to emphasize that an unexpected event may or may not cause an exception, monitoring resources, supply process orders and detecting deviations in their planned values is the way to detect unexpected events, the reference model intrinsically implies methods to detect unexpected events if there is a monitoring system of the status of resources (their availability) and the status of scheduled supply process orders (their specification) but this is not in the scope of this work in which the unexpected event is given. In fact as far as Supply Chain Execution Control concerns the only possible exceptions are caused by unexpected events that in an utopist Supply Chain can always be tracked to the unavailability of one resource, but in realistic terms resource descriptions when building execution schedules are incomplete and whenever two different business partners interact they might have an incomplete view of each other's resources. So for these two latter cases, unexpected events are associated to supply process orders.

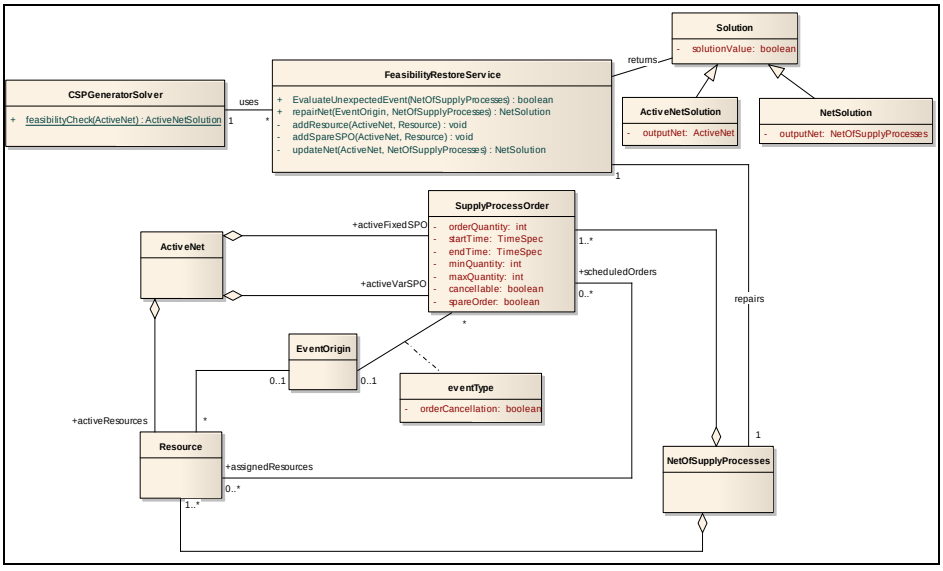

Fig. 2. Repair Service Class Diagram

4.2 The Feasibility Restoration Service internal mechanism.

The main functionality of the FRS is providing the functions of evaluating the effects of an unexpected event and repairing a disrupted schedule. Following we give a brief description of algorithm taking place when searching for a feasible solution.

Unexpected events are of two different types:

1. Events produced by unexpected changes in the availability of a controlled resource.

2. Events produced by unforeseen changes in the requirements of a Supply Process Order.

Events can also be produced by simultaneous changes in the availability of several resources and/or in the requirements of a several supply process orders. In this case, the initial reparation set will be composed of a set of resources and related orders.

An unexpected event may not cause an exception because planned resource and supply process orders slacks (in form of hedge capacity, inventory buffers, variable order sizes) might absorb automatically the variation caused by the event. Or in fact as the following mechanism proposes making small changes in resource utilization and supply process orders, reallocating these slacks can prevent the exception from taking place.

Suppose an event affected a Resource; the first stage of the mechanism checks the ability of the affected Resource to fulfill its scheduledOrders exactly as they were planned. Along the execution of this mechanism the set of SupplyProcessOrders which join it and keep their original specification is called activeFixedSPO.

If the event does not compromise the execution of any SupplyProcessOrder, which means that the CSP associated with the activeNet is feasible, then the event does not generate an exception and the outcome of the mechanism a feasibility confirmation. If a feasible solution to the activeNet's associated CSP was not found, the mechanism advances to the next stage, expanding the activeNet, and this stage is repeated until the stopping condition is fulfilled. This expansion consists in:

First: include in the set of activeVarSPO (the set of SupplyProcessOrders subject to modifications in the CSP), every spareOrder belonging to the set of scheduledOrders in the activeResources set (the set of resources in the activeNet). This also implies including the assignedResources of these recently included spareOrders, together with the scheduledOrders of these resources in the set of activeFixedOrders. If this new activeNet is feasible the Net of SupplyProcessOrders is updated with the operation updateNet, and the mechanism finishes, otherwise:

Second: Expand the net, making every SupplyProcessOrder in the activeFixedSPO set variable by including them in the activeVarSPO set and including all their assignedResources in the mechanism, together with their scheduledOrders in the set activeFixedSPO.

The stopping condition of the mechanism is fulfilled whenever a feasible solution to the CSP associated to the activeNet is found or there are no possibilities of expansion, that is the set of activeFixedSPO is empty.

The design of the Repair Mechanism was based in the concept of minimal impact on the current execution schedule, assuming that: modifications to SupplyProcessOrders and the use of resources is limited to their planned slacks, and that SupplyProcessOrders and Resources involved first are the nearest to the event origin, and expanding the involved elements only if necessary. This is a gradual radial expansion, as shown in (Fig. 3). In the figure, as the expansion takes place, the involved Resources and SupplyProcessOrders conform the ActiveNet (marked with V variable SPOs, and with F fixed SPOs) which is the subset of the Net under repair where feasibility is searched by means of the Constraint Satisfaction Model for Feasibility Check and Restoration.

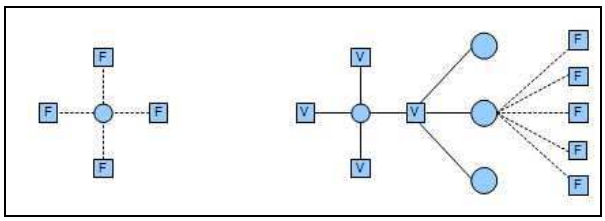

Fig. 3. Radial Expansion of the ActiveNet

Every expansion requires a request for Schedule information as stated in 3.1.

5 Urea Supply Chain Case Study for empirical validation

This Case Study assumes that within the supply chain exists an operative Supply Chain Execution Control System (with the functions of monitoring and requesting for the intervention of the Repair Service) where the control actions (monitor and detect exceptions) are generated using the reference model presented in section 2, therefore execution plans are described using the same reference model, and exception detection and repair actions are fulfilled using the Repair Mechanism.
In this Supply Chain Urea is produced, warehoused and distributed to three geographically distant Distribution Centers (DCs). The Factory Warehouse is located in Bahia Blanca (FWBahiaBlanca), Argentina, and DCs are: "Urea-DCSanLorenzo" at San Lorenzo, Argentina; "Urea-DCUruguay ", at Montevideo, Uruguay; "Urea-DCBrasil", at Rio Grande, Brasil.

A Distribution Resource Planning (DRP) System is used to generate a distribution schedule for a scheduling horizon of 33 days. In this schedule, product (Urea) availability in FWBahiaBlanca is considered to be infinite, this means that urea stock, demand and supply is managed for each Distribution Center attending constraints regarding to: Ships routes and availability, loading dock availability at factory warehouse, and inventory size and safety stocks constraints at each DC.

Two unexpected events were simulated the first consisted in the unavailability of a ship used to source "DC-Brasil" which was solved by using an alternative resource, provided by a 3PL (3rd party logistics provider). And the second consisted in a set of concurrent events that implied that 10 orders from DC-Uruguay augmented their size. This event was caused by absorbing the market share of a competitor who had an inventory stock-out. This would have caused an exception but the FRS found a solution and the possible exception and the solution can be seen in (Fig. 5). The solution consisted in checking the possibility of making an earlier second replenishment of Urea, this implied checking capacities and states of the ship, the loading dock, and the availability of Urea.

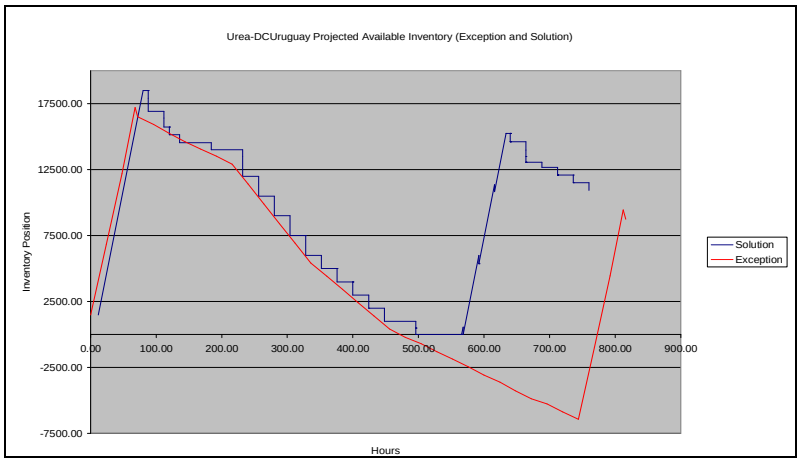

Fig. 4. Exception Solution given by the FRS

6 Conclusions and Future Work

In this contribution we have proposed a foundation for the development of Service Oriented Autonomous Exception Management Systems. The problem of semantic interoperability between Supply Chain partners can be completely solved by defining business documents (portions of an Execution Schedule) based on the Exception Management Reference Model, in fact, an instance of this model constitutes a full execution schedule with the necessary data for feasibility evaluation and restoration.

At the same time the business process of Exception Management has been described as a BPMN model, allowing transformation to executable languages such BPEL. The EMBP has two actors the SCEC and the FRS, so for the last we gave a description of the internal mechanism. This FRS provides local and within slacks solutions for any disrupted schedule. The only constraint is that the Schedule has to be described using the Exception Management Reference Model. Schedules might be of any nature from floor plant schedules to distribution schedules, and might consider a full Supply Chain execution schedule.

Based in this contribution two working lines arise, the first is implementing a FRS, which is a complex task because requires the interaction of some specifically designed algorithms to solve the corresponding CSP of the active net.

And the other working line is extending the EMBP to support the utilization of the FRS in a coordinated fashion by different Supply Chain partners that can collaborate in order to mitigate the effects of an extended Exception. The latter implies the design of another Web Service for the coordination of concurrent Supply Chain Execution Schedules, which are linked by relatedOrders as in 3.1.

References

1. Radjou, N., Orlov, L.M., Nakashima, T.: Adapting To Supply Network Change. (2002)
2. András Pfeiffera , Botond Kádára , Monostoria, L.: Stability-oriented evaluation of rescheduling strategies, by using simulation. Computers in Industry 58 pp. 630-643 (2007)
3. Arief Adhitya, R.S., I.A. Karimi: A model-based rescheduling framework for managing abnormal supply chain events. Computers and Chemical Engineering (2006)
4. Bing Wang, Liu, T.: Rolling Partial Rescheduling with Efficiency and Stability Based on Local Search Algorithm. In: Conference Rolling Partial Rescheduling with Efficiency and Stability Based on Local Search Algorithm. (2006)
5. A. Guarnaschelli, O.C., E. Salomone: Sytematic Repair of Disrupted Supply Processes. In: Conference Sytematic Repair of Disrupted Supply Processes. (2008)
6. Song, Y., Han, D.: Exception specification and handling in workflow systems. Proceedings of the 5th Asia-Pacific web conference on Web technologies and applications, pp. 495-506. Springer-Verlag, Xian, China (2003)
7. Yuan, H.-t., Ding, B., Sun, Z.-x.: Workflow Exception Forecasting Method Based on SVM Theory. In: Conference Workflow Exception Forecasting Method Based on SVM Theory, pp. 81-86. (2008)
8. Hwang, S.-Y., Tang, J.: Consulting past exceptions to facilitate workflow exception handling. Decis. Support Syst. 37, 49-69 (2004)
9. Weske, M.: Business Process Management: Concepts, Languages, Architectures. Springer-Verlag New York, Inc. (2007)
10. Hamadi, R., Benatallah, B., Medjahed, B.: Self-adapting recovery nets for policy-driven exception handling in business processes. Distributed and Parallel Databases 23, 1-44 (2008)
11. Griffith, R., Kaiser, G., L, J.A., 243, Multi-perspective evaluation of self-healing systems using simple probabilistic models. Proceedings of the 6th international conference on Autonomic computing, pp. 59-60. ACM, Barcelona, Spain (2009)
12. Friedrich, G., Fugini, M., Mussi, E., Pernici, B., Tagni, G.: Exception Handling for Repair in Service-Based Processes. Software Engineering, IEEE Transactions on 36, 198-215 (2010)
13. Bearzotti, L., Salomone, E., Chiotti, O.: An autonomous multi-agent approach to supply chain event management. In: Conference An autonomous multi-agent approach to supply chain event management, pp. 524-529. (2008)
14. Montgomery, N., Waheed, R.: Supply Chain Event Management Enables Companies to Take Control of Extended Supply Chains. (2001)
15. Zimmermann, R.: Agent-based Supply Network Event Management (2006)
16. Cauvin, A.C.A., Ferrarini, A.F.A., Tranvouez, E.T.E.: Disruption management in distributed enterprises: A multi-agent modelling and simulation of cooperative recovery behaviours. International Journal of Production Economics 122, 429-439 (2009)
17. Tania BedraxWeiss, Conor McGann, Ramakrishnan, S.: Formalizing Resources for Planning. In: Conference Formalizing Resources for Planning. (2003)
18. Le Pape, C.: Implementation of Resource Constraints in ILOG Schedule: A Library for the Development of Constraint-Based Scheduling systems. Intelligent Systems Engineering 3, 55 - 66 (1994)
19. OMG: OMG/BPMN 1.2: OMG Specification (2009)

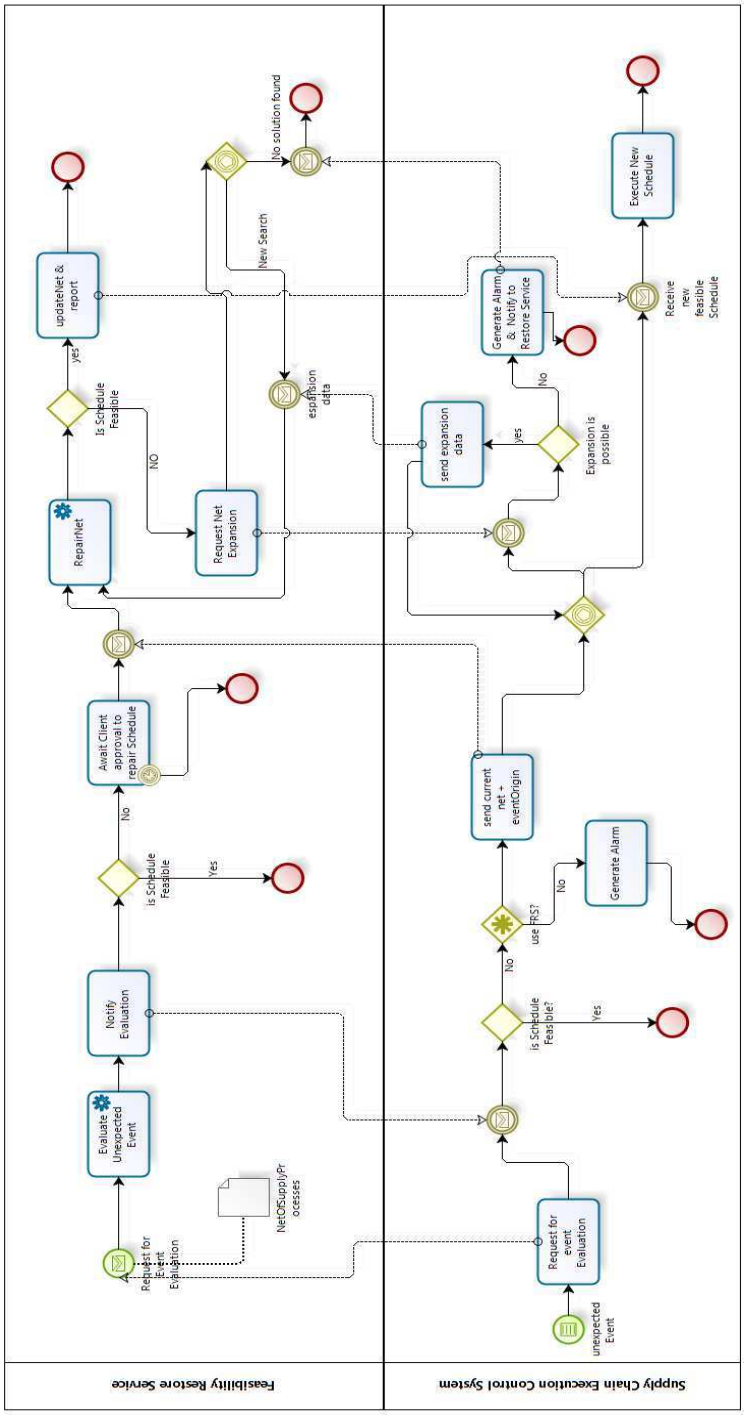

Fig. 5. Exception Management Business Process