Computers in Industry 80 (2016) 30–42

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.elsevier.com/locate/compind

Framework for modelling and simulating the supply process monitoring to detect and predict disruptive events

Erica Fernández[a,b,*], Verónica Bogado[c], Enrique Salomone[d], Omar Chiotti[d,e]

[a] Departamento Ingeniería Industrial, FRRA, UTN, M. Acuña 49, Rafaela 5900, Argentina
[b] GEMPRO – UTN FRSF, Lavaisse 610, Santa Fe 3000, Argentina
[c] Departamento Ingeniería en Sistemas de Información, FRVM, UTN, Av. Universidad 450, Villa María 5900, Argentina
[d] INGAR – CONICET, Avellaneda 3657, Santa Fe 3000, Argentina
[e] CIDISI – UTN FRSF, Lavaisse 610, Santa Fe 3000, Argentina

ARTICLE INFO

Article history:
Received 15 December 2015
Accepted 11 April 2016
Available online 26 April 2016

Keywords:
SCEM system
Monitoring system
Disruptive event
Simulation

ABSTRACT

Disruptive events that take place during supply process execution produce negative effects that propagate throughout a supply chain. Event management systems for supply chains have emerged to provide functionality for monitoring schedules, managing disruption, and repairing schedules affected by a disruptive event. A Web service that provides a schedule monitoring functionality for supply chain event management was developed. This paper provides a framework to allow enterprises that hire this service to develop simulation models of monitoring processes and evaluate their ability to detect and anticipate disruptive events. The framework, based on discrete event simulation, is implemented in a library that can be used for developing and testing monitoring processes by means of a friendly interface. A marine freight transport process was used as a case study to show how a supply process and its environment can be modelled and simulated by using the library. Simulation results show the ability of this approach to anticipate disruptive events and identify critical stages of a supply process in order to prevent disruptive events.

1. Introduction

In an integrated supply chain, the overall performance largely depends on keeping the coordination of schedules for producing and distributing goods. These schedules are generated by the planning subsystem of the Enterprise Resource Planning (ERP) system of each enterprise and are typically represented by production and distribution orders, where each order represents a particular instance of a generic supply process.

During the execution of scheduled orders, significant changes may occur either in the specification of orders or in the availability of involved resources. These unplanned changes, called disruptive events, can produce negative effects that propagate throughout the supply chain, affecting schedules and their coordination [1,2].

The paradigm of robust planning proposes to define buffers (material, capacity, and time) to absorb changes that may occur during the execution of scheduled orders [3]. These buffers allow achieving a schedule most likely to remain stable during its execution. This avoids re-planning tasks, which can be costly and time-consuming, since all enterprises involved in the supply chain should agree on a new collaborative plan. However, buffers cannot usually absorb all changes due to the impossibility of forecasting with certainty the time and place in which disruptive events could occur and their magnitude.

Under this scenario, Supply Chain Event Management (SCEM) Systems have emerged [4,5]. SCEM systems should provide functionality for: *monitoring schedules* during its execution to detect disruptive events (*reactive monitoring*) or to prevent disruptions before they occur (*predictive monitoring*); *managing disruption* after a disruptive event to check if schedules are still feasible; and *repairing schedules* affected by a disruptive event considering the distributed nature of a supply chain.

Fernández et al. [6], present an Agent-based Monitoring Service for Management of Disruptive Events in Supply Chains (MSMDE), which is a Web service that provides the *schedule monitoring* functionality of a SCEM system named collaborative management of disruptive events in supply chains presented in [7]. Enterprises that hire monitoring Web service MSMDE must provide a monitoring model of the supply process to be monitored. For

* Corresponding author at: Departamento Ingeniería Industrial, FRRA, UTN, M. Acuña 49, Rafaela 5900, Argentina.
E-mail addresses: ericafernandez@santafe-conicet.gov.ar (E. Fernández), vbogado@frvm.utn.edu.ar (V. Bogado), salomone@santafe-conicet.gov.ar (E. Salomone), chiotti@santafe-conicet.gov.ar (O. Chiotti).

http://dx.doi.org/10.1016/j.compind.2016.04.002
0166-3615/

that, enterprises must develop the monitoring model by using the abstract modelling language provided by the reference model for supply process monitoring presented in [8]. This reference model is a meta-model that specifies the abstract syntax of a modelling language to represent the static part of the monitoring model of a supply process [8]. The monitoring model represented in terms of the reference model is automatically transformed by the Web service MSMDE into a monitoring process, which is used by the service for monitoring the supply process.

So, enterprises that hire this Web service must understand the abstract modelling language provided by the reference model, which is not easy to understand and use. As Moody states [9] using an abstract language for model building and testing may be a difficult task to perform and prone to mistake if a suitable tool is not available.

The objective of this paper is to provide a framework to allow enterprises that hire Web service MSMDE to develop simulation models of monitoring processes and evaluate their ability to detect and anticipate disruptive events without the need of knowing the abstract modelling language provided by the reference model. The framework, based on discrete event simulation, is implemented in a library that contains a set of simulation elements related to concepts of the reference model. The library can be used for developing monitoring processes through a friendly interface, hiding the abstract modelling language provided by the reference model. Different monitoring models tailored to supply processes can be developed and tested by setting input parameters.

The remainder of this paper is structured as follows: Section 2 discusses related works. Section 3 briefly defines the concepts of the reference model. Section 4 presents the framework and its implementation in a library. Section 5 describes a case study and Section 6 presents conclusions and future work.

2. Related works

2.1. Approaches for supply process monitoring

Approaches for reactive monitoring of schedules are based on capturing information about material resources and/or order specifications during execution for assessing performance indicators or rule-based arithmetic ratios in order to detect disruptive events. Indicators are also used to assess the impact of disruption along the supply chain. Among the proposals for reactive monitoring, those presented by Bansal et al. [10], Liu et al. [11], and Winkelmann et al. [12] can be mentioned.

The approach presented by Bansal et al. [10] uses performance indicators assessed at regular intervals, cause-effect relationship models to identify the root cause of a disruptive event and rectification strategies to repair the schedule. Liu et al. [11] present a methodology that uses Petri nets to formulate supply chain event rules and analyse cause-effect relationships among events. Events are classified in the following types: task status-related events, events produced by a task, and external events. Based on interactions between partners in the supply chain, events are identified and rules relating them are defined to represent a supply chain in order to propagate events, analyse them, and suggest a solution when a disruption is detected. Winkelmann et al. [12] present an approach for conceptual modelling of SCEM systems. The main tasks are supply chain process definition and identification of relevant logistical objects where disruptive events could occur. These objects are used as reference points for monitoring activities. For each object, arithmetic ratios are defined and combined with a rule-based expression to detect disruptive events. Once detected, disruptions are notified to applications or people for corrective actions.

The main technologies used for tracking and tracing orders and/or resources during the schedule execution are RFID and GPS [13]. RFID is generally used for monitoring products, pallets, and machines, and GPS for vehicle location. The massive data stream coming from a supply chain when each object provides its current status requires the use of data flow processing technologies. The main technologies commonly used for this purpose are: complex processing [14], agents [15], and event-condition-action [16]. A complex processing technology is a pattern-based event processing. Relevant events are selected by data stream filtering and matched to predefined patterns to derive complex events that allow detecting disruptive events. Agents' technology is a software application including complex algorithms able to collect up-to-date data about orders and/or resources and filter them for capturing relevant information, and with knowledge and reasoning capability for identifying disruptive events. The event-condition-action technology extends traditional databases with a layer of rules and event detection mechanisms for event processing. Among the proposals based on data flow processing technologies, those presented by Meyer et al. [15], and Ko et al. [17] should be mentioned. Meyer et al. [15] present an agent-based architecture for collecting up-to-date information of products and comparing their current status with that planned in orders so as detect disruptive events. Agents determine the planned status of products by analysing the schedule information (such as order due dates and planned transactions and operations that will affect the product). If a disruption is captured, agents propose solutions or suggest how to reduce the severity of the problem. Ko et al. [17], present an agent-based system for monitoring product locations. A monitoring agent checks product arrivals at nodes specified in the monitoring plan. If product arrivals are detected within the planned time period, the monitoring agent visits the next node. Otherwise, it suspends monitoring and searches for products deviated from their planned path.

Approaches for predictive monitoring of schedules are based on capturing information about resources, order specifications, and environment variables during execution to prevent disruptions before they occur. Some proposals for predictive monitoring should be mentioned: those presented by Kim et al. [16], Fernández et al. [8], and Vlachakis and Apostolou [18]. Kim et al. [16] propose a rule-based language to develop monitoring models to predict and prevent business process disruptions before they occur or detect and repair them once occurred. The rules defined to capture information of business processes have an event-condition-action structure extended with other components such as contexts in which rules are applicable, preferences that specify rule priorities to be triggered by events, and frequency that specifies periods in which rules are checked. Fernández et al. [8], present an abstract language to develop monitoring models of supply processes. It is based on cause-effect relationships among variables that represent features about order specifications, resources, and the supply process environment. These variables are monitored in different milestones related to supply process stages to anticipate or detect disruptive events. To this aim, predictive or reactive evaluation functions must be defined. These can be simple mathematical functions or more complex ones such as Bayesian Networks, Petri Nets, Complex Processing, or rule-based. Vlachakis and Apostolou [18] define a supply chain event management framework able to capture events related to resources, orders, or environment and process them to detect undesired deviations when the predefined threshold is exceeded. After a disruptive event is detected, the damaged schedule is repaired according to predefined rules or decision models.

Languages to develop models for supply process monitoring in the supply chain context as that proposed by Kim et al. [16] and Fernández et al. [8] have the advantage of an abstract syntax that