Corporate Logistics Improvement through Automated Processes: A Romanian Case Study for a Repair Flow Control and Supply Management System

Mihai Orzan, Academy of Economic Studies, Bucharest, Romania, mihai.orzan@ase.ro
Liviu Serbanescu, Romanian Astronomic Academic Institute, Bucharest, Romania, liviu_969@yahoo.com
Gheorghe Orzan, Academy of Economic Studies, Bucharest, Romania, orzang@ase.ro
Violeta Sima, Petroleum-Gas University, Ploiesti, Romania, violeta.sima@yahoo.com
Ileana Georgiana Gheorghe, Petroleum-Gas University, Ploiesti, Romania, ileghe@gmail.com

Abstract

Manufacturing systems are often costly to develop, operate, and maintain. Their flawless functioning is a major premise of market success in today's global market. In this article we describe a complex supply management and repair flow control system, developed in order to enhance the competitive power of a company in dynamic environments. The system is used by the Romanian branch of a multinational communications provider (focused mainly on GSM) and functions using the organizational Intranet (we are about to transform it in an Extranet, involving some strategic suppliers of the organization). It was developed as a Repair Flow Control system, but it necessarily evolved to cover all supply management activities associated with technical support of current organizational activities. It also includes a complex, automated fuzzy rule set system, used for decision support and independent system control and management.

1. Introduction

Supply chain management is one of the most important topics in the field of logistics, as well as in the broader topics of marketing and information systems, with increased relevance, both for industry and the society as a whole. Advances in information technology have enabled fast and reliable communication among different links placed along the supply chain, which lead to a smooth flow of materials, parts, and information from producers to users, in a very efficient manner [1]. However, due to increase complexity in modern production and commercial systems, slight errors can bring the whole system crashing down, with tremendous detrimental effects for an organization's stakeholders. On the other hand, efficient supply chain system can improve a product time to market, eliminate productive downtimes (due to out-of-stock situations), increase profit by reducing inventory as well as cost among others, and ultimately to increase customer satisfaction [2].

Managing a supply chain involves the flow of resources across the entire supply chain. A supply chain could contain anywhere from 2 through several stages (e.g., manufacturers, suppliers, assembly, retailers) and each of these stages could possibly involve several entities that either compete or cooperate to provide the service, product, materials requested from another stage in the chain [3].

One of the traditional methods of ensuring smooth and continuous production is the maintenance of inventory for the resources used throughout the productive and support processes. Inventory usually allows an organization to maintaining independence among operations, meeting variation in demand from downstream and supply from upstream, and taking advantage of market opportunities. In an ideal world, at any given point in time, each of the entities at different stages in the supply chain has just enough stocks to fulfill every order that comes in. Achieving this is rather straight-forward when there is no variability in order type, quantity, as well as frequency, which, of course, is a scenario rarely encountered in most real-world situations. Therefore, it is necessary to be able to dynamically deal with variations in the environment (e.g., demand, supply, or monetary constraints) to avoid both stock-outs as well as over-stock problems [4].

2. Conceptual Background

One of the most important aspects of supply chain management is related to the administration and control of internal stocks, a concept known as inventory management [5]. Actors in supply networks have to find a balance between keeping inventory, with its accompanying inventory costs, and fulfilling the demand of their customers. When structurally keeping an inventory that is larger than required, the costs will reduce the profit. However, when customers are faced with a depleted supplier too often, they can lose confidence and switch to a competitor, thereby decreasing the total sales of the actor, and thus the profit. Several heuristics and methods exist, including look-up tables, but with limited detailed modeling [6].

One of the earliest works in this field [7] proposed a deterministic multiproduct, multi-facility production, and inventory model focused on determining an optimal production schedule, which specifies how much each facility in a networked corporation should produce so that the total cost is minimized. [8] developed a model structure that can be used to predict the performance of a firm with respect to production costs, service quality, responsiveness and flexibility of the production/distribution system, in an integrated, hierarchical, and stochastic network model structure. Another interesting paper [9] presents a generic model of a stock management model consists that of two parts, the physical stock and flow structure of the system, and the decision rules used to control the system. Further factors to include in this hypothesis of decision-making are the

availability, timeliness and perceived accuracy of information regarding the supply line, while [10] presented a case study about linear programming optimization of a network for an aluminum plant.
The mathematical structure of their problem is a generalized network with constraints such as capacity constraints and flow constraints between each node. The concept of Quick Response systems [11] is another typical research subject associated with the development of inventory management policies, aiming to allow faster response to market demand changes whilst maintaining lower inventory levels.

In later years the research implied more and more the notion of an intermediary agent for the inventory management system, an approach that can be classified according to their system architecture: the Hierarchical approach, the Federation approach and the Autonomous agent approach [12, 13]. A combination of hierarchical and semiautonomous agents is used in typical modern manufacturing enterprises. [14] introduces the concept of facilitator agents, which help the other agents in communicating with each other. The main function of a facilitator is to route outgoing messages to the correct destinations and to translate the incoming messages for consumption by its agents.

Most inventory management studies assume a single central decision maker. However, in real life many supply chains are associated with several companies, each with their own goals and objectives, [15] being one of the first studies that have included cases where decentralized control is allowed. Another important problem when dealing with inventory management is the so-called safety stock size, which determines a flag value of the stock for each inventory product, which will trigger the launch of an order to the supplier. In this case we use the classic approach [16], called {r,Q}, where Q is the quantity ordered when the internal inventory hits r units. In this case, the safety stock is determined for a given level of spare parts usage (lifetime usage, according to manufacturer specifications), where lead times and demands are stochastic.

We have improved the traditional supply chain approach with radio-frequency identification (RFID) tags that allow an efficient, real-time tracking of all spare parts in the system. RFID system have become increasingly being used in everyday scenarios, ranging from inventory control and tracking, to medical-patient management [17]. The key driver behind this widespread adoption is the simplicity of the tags, which enables very low (nearly zero) cost at high volumes. The tags themselves vary significantly in their capabilities, from dumb tags which merely transmit a particular bit-string when probed by a reader, to smart tags which have their own CPU, memory and power supply. Most of these tags are designed to have a very long life, and hence do not use any existing energy sources for transmitting data [18]. Rather, they derive the energy needed for transmission from a probe signal sent by a reader node. By reading all the tag IDs in the neighborhood and then consulting an internal database that provides a mapping between IDs and objects, the reader learns the existence of the corresponding objects in the neighborhood.

3. Automated Inventory Management with RFID Support

The system we have developed, called STOKY_RFC (Repair Flow Control) was created in order to ensure real time flow control of on-site equipments for large technical facilities and was implemented in one of Romania's largest GSM operators (part of a multinational communications company). It is aimed mainly at automated inventory management and automated maintenance activities (scheduled or on-demand repairs). It is also used to monitor supply chain management and repairs, either on-site or with strategic suppliers. The model we considered in this paper assumes just a part of a much larger supply chain system, with only two stages: spare parts suppliers and inventories maintained by the company [19]. The spare parts can be in use, in which case each one has scheduled maintenance operations associated, they can be sent for repairs (either on-site or off-site) or they can be placed in stock, for later use. The whereabouts of each item is determined through the use of passive RFID tags.

The STOKY_RFC system provided a number of options to integrate with a wireless sensor platform. We chose to use a RS232 connection as the point of data communication since the wireless sensor platform offered a UART (Universal Asynchronous Receiver Transmitter) at the same bit rate. An additional circuit was introduced to interface the UART of the wireless sensor platform to the RS232 connection of the RFID reader to correctly convert the transistor-transistor logic (TTL) levels.

Its architecture has a central module, used to manage groups of sites, called a ***COM*** unit, while several ***CES*** modules supervise the whereabouts and states of equipments. COMs also manage the supply chain of components and spare parts.

The system manages two different groups of modules: **Editing** and **Reporting**. The Editing Module performs the following functions:

1. ***Reception*** – this module is used for data collection, mainly input data for the COM/CES deposits and they come from:
   - directly from the operator (new inputs in COM/CES deposits);
   - from other COM or CES modules – these are collected using a query ("polling") system that uses a transfer zone (buffer) on the server. The data resulted from other COM/CES

deliveries by "polling" may be rejected by users back to the transfer buffer on the server, and the **sender module** is notified about this;
- from "Repair3000" module for repaired equipments;

2. ***Delivery* -** this module allows:
   - delivery of equipments to other COM/CES, suppliers or on-site;
   - the return of rejected data by other COM/CES in the enter module;
   - to edit "delivery form";
3. ***Replacements & transfers*** –this module allows the system:
   - to undergo direct replacements of stocks;
   - to transfer the equipment from one location to another;
   - to change the state of the equipment.
4. ***Repair3000*** *(edit)* – is a module used:
   - to lift equipments (from own inventory, either on-site or from a different location, as well as from strategic suppliers) for repairing;
   - to edit data from **R3000/FRD Forms** and print them;
   - to make entries in the Equipment Repair Form;
   - to manage the physical equipments replacement.

The relations between the modules are presented in Figure 1.

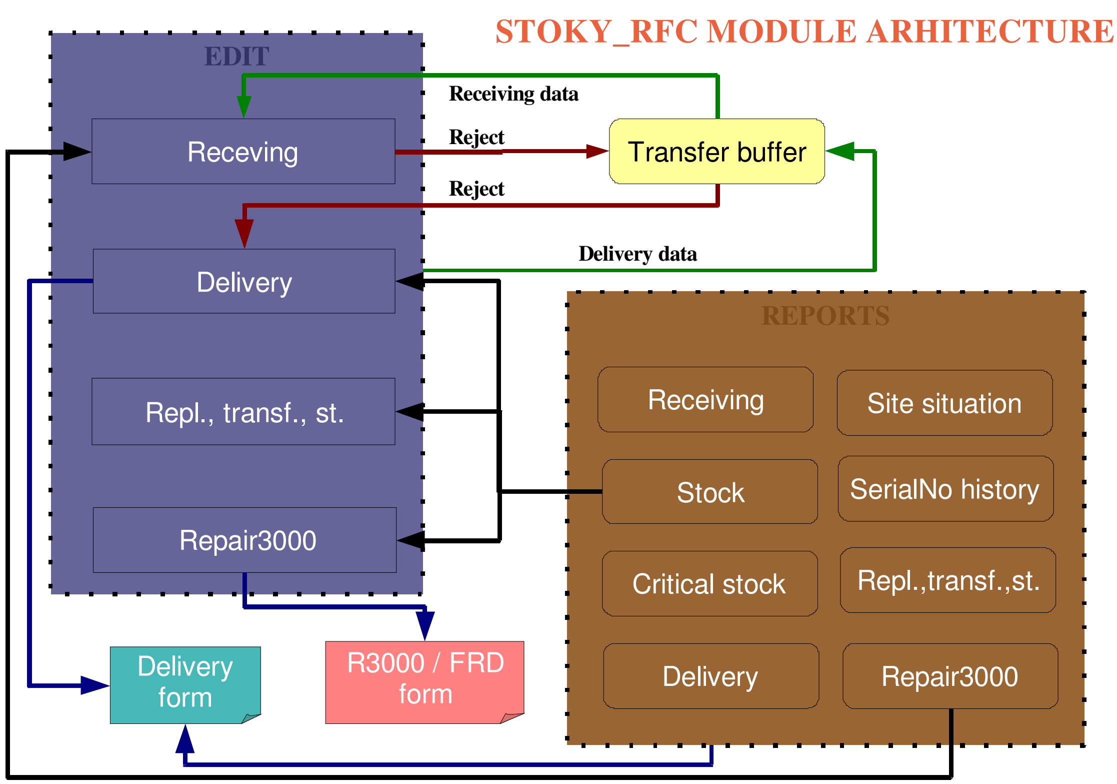

Fig.1. STOKY_RFC Module Architecture

The Reporting module has the following functions:

1. ***Reception*** –this function allows the system to report inputs to the COM/CES deposits;
2. ***Stock*** – this module is used to report about inventory levels of the COM/CES deposits;
3. ***Critical stock*** – this function allows users to set alarms on critical inventory levels, as well as make exceptional and regular reports with regard to minimal inventory levels;
4. ***Delivery*** – this module produces reports about the outputs of the COM/CES deposits, as well as faulty on-site equipments.
5. ***Site Status*** allows the consultation of:
   - on-site entry data;
   - on-sites output data;
   - input/output differences and on-site equipment replacements.
6. ***Serial No History*** – this function is used to deliver reports about the history of product lines (stock/site entries, stock/ site outputs, and replacements);
7. ***Replacements & Transfers*** – used to report on: direct lines replacements, transfer between locations, as well as equipment state changes.
8. ***Repair3000*** *(consult)* module delivers reports about the state of equipments that are

being to be send for repair (outgoing transports), are being repaired, or are coming back from repairs (incoming transports).

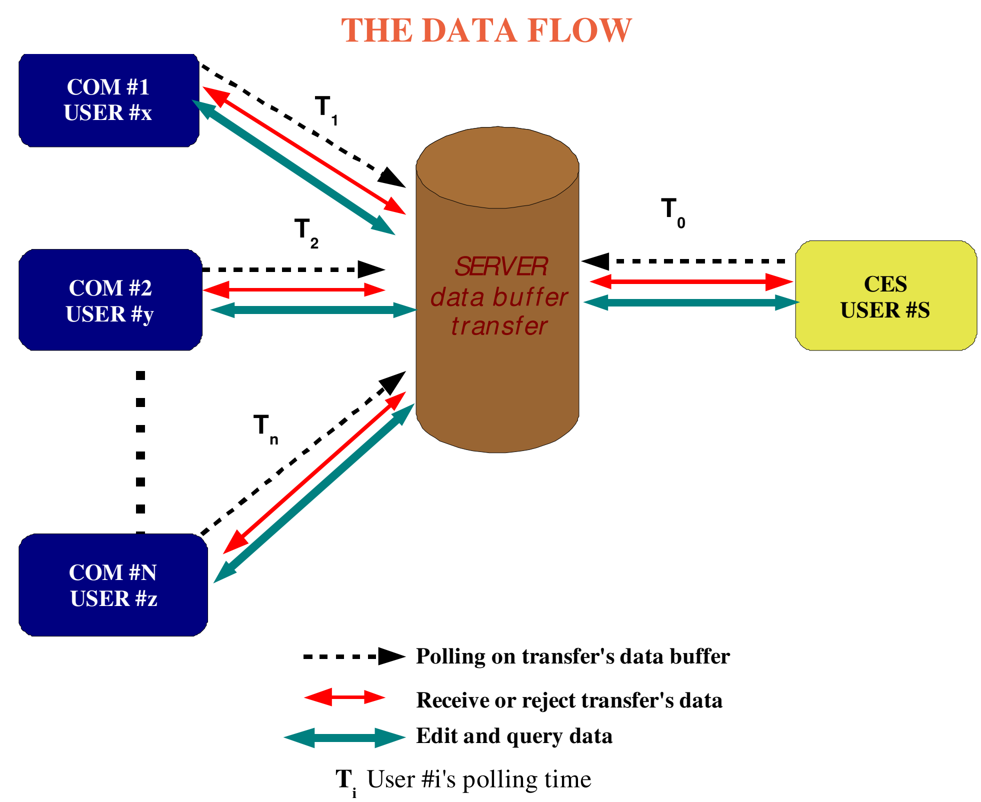

Fig.2. STOKY_RFC Data Flow

Based on data gathered from all these sources users can take decisions regarding shipment size the sizing of the groups with spare parts (or STOKY_RFC can take these decisions for them, automatically). This can be done by a fuzzy rules system.

For real time functions the system includes a periodical transfer buffer query system (polling), its frequency established by users. Usually, both COM and CES have many different human online users simultaneously, that use same functions in same or different modules, which imposes a high system flexibility for STOKY_RFC. The users are allowed to edit their own forms and reports, using uniquely system generated access credentials. The system also allows for anonymous users, but with limited access, generally to specific basic Reporting functions on the COM module.

All operations run directly on the system server, which facilitates control function for the entire system. The STOKY_RFC data flow is included in figure 2. Outputs can be directed to different COMs orCESs, strategic partners (supplier), using an Extranet system, or on-site. COM and CES outputs can be rejected either at inventory input form (BL) level, or at item (equipment) level.

The data flow detailed in Figure 3 and can be done in one of the following fashions:

- By Sender - until the outgoing validation on BL level;
- By Receiver - until input validation on the BL level – this can lead to the cancellation of transmission and the message will reentry in the "polling" system, and from here it can be rejected to the sender. Rejected inventory lines can be retransmitted or cancelled.

Outgoing BLs can be found in one of the following states:

- Preparation phase – gathered from inventory data, which are reserved in a quantitative manner;
- Transfer phase – BLs that are being sent to other COMs or CESs, which have been validated by the Sender;
- Rejected – include all BLs rejected by the Sender (COM/CES modules).

In order to expedite the maintenance management and control, the system includes a set of individual equipment states. It also has an explicit state transition system, which describes all corresponding changes from one state (phase) to another. These state transitions are detailed in Figure 4.

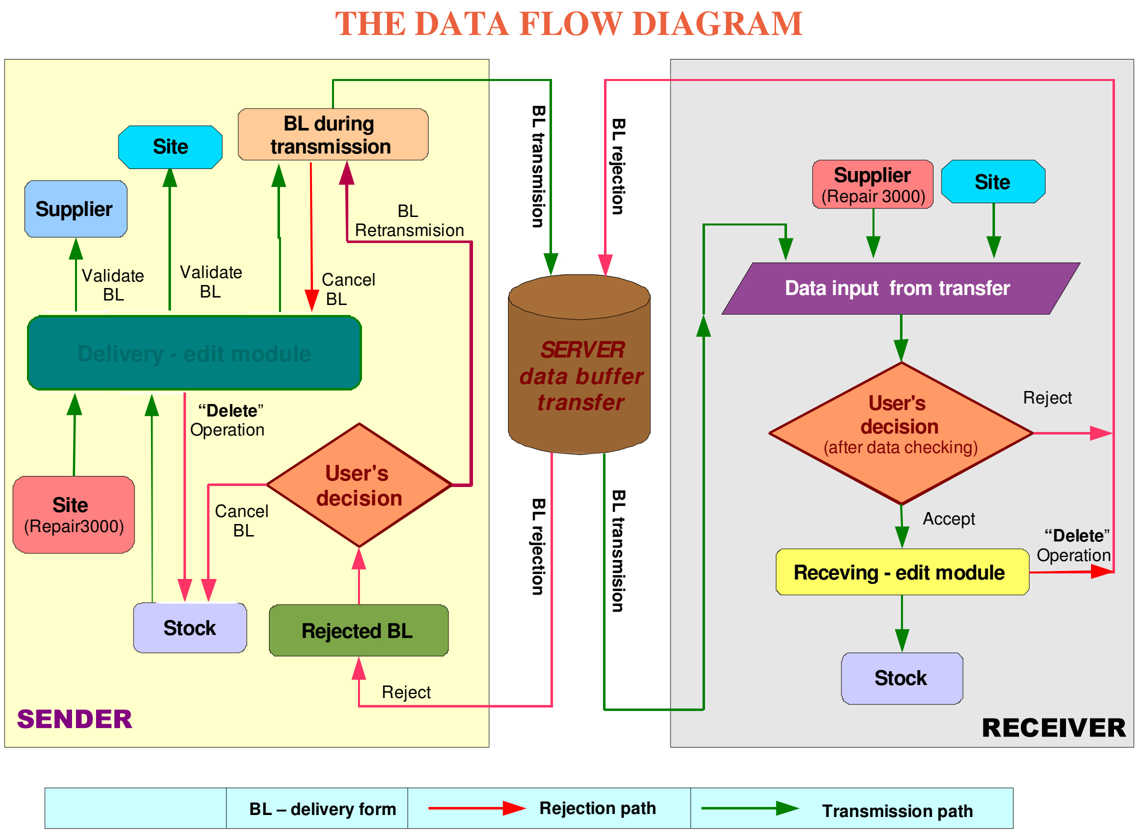

Fig.3. STOKY_RFC Data Flow Diagram

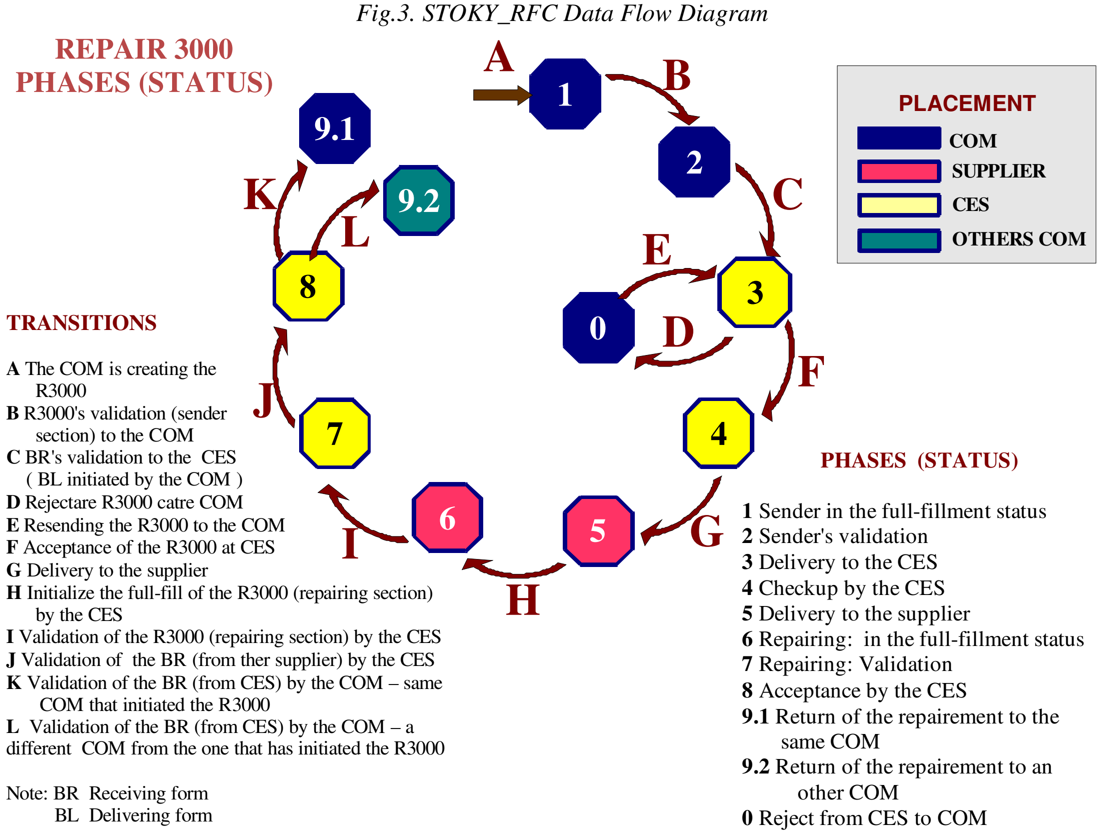

Fig.4. STOKY_RFC States & Transitions

The Input Edit Module of COM/CES inventory is based on development and validation of Inventory Input Forms (BL). Input data can come:

- direct – complete operator (human) entries;
- from "polling" – collected via the "polling" system – one item can be taken from an Inventory Input Form (BL), or all its items can be received, as a whole.

This can be done in the selected BL or different, older BLs;

- from on-site current activities;
- from a strategic supplier (when an item enters the stock or when it is returning from repairs) – in this case the BLs are collected using selections of *Repair3000* query module.

The input data (before the validation of the Inventory Input Forms) can be cancelled, in which case it reenters the pooling system (this is detailed in figure 5). The validation of Inventory Input Forms (BLs) leads inevitably to stock data transfers.

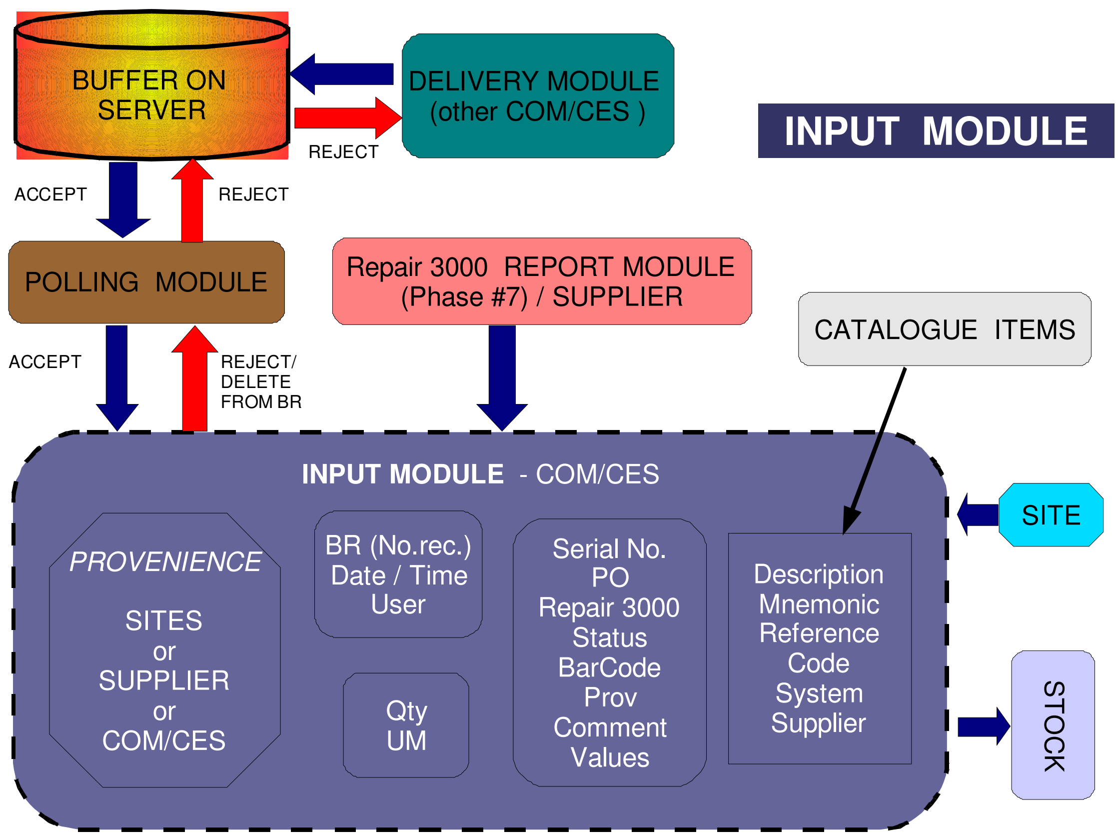

Fig.5. STOKY_RFC Input Module and its components

Deliveries are pooled in outgoing BLs which are sent to other COM/CES via the "pooling" system, to strategic partners as well as to other company sites.

Repair3000 module also allows reporting through the *Delivery For*" function, implemented according to Figure 6. In addition, the *Repair3000* module also manages and controls equipment repair activities (detailed in Figure 7), which includes:

- information gathering from the repairs queue and repair reporting using the *Repair3000* form or a FRD (Fault Report Data) Form;
- coordination and implementation of replacements supply chain activities for damaged and faulty equipments (usually, these equipments can be replaced with similar ones on stock, with similar equipments on different company sites or with replacement products from suppliers);
- Label and directory generation, reporting and printing in any user personalized formats.

The decision regarding the spare parts group size is automated using a table with fuzzy type rules. This has the following parameters (Figure 8): ***ND*** (the nature of the defects), ***FD*** (frequency of defect occurrences), ***CD*** (defect causes), ***FCD*** (occurrence frequency for each defect cause), ***MTBF*** (Mean Time Between Failures – a numeric value, provided by each product's supplier), ***TI*** (intervention time for maintenance and/or repairs), ***PG*** (geographic position), ***IP*** (relative importance of a spare part for the company), ***IL*** (work point relative importance for the company).

After identification of these parameters we determine intermediary relations like: FD (ND, FCD), MTBF (FD), and so on. Each variable has a specific weight denominated $\mathbf{P_i}$, where **i** is each of the following: {ND, FD, FCD, MTBF, TI, PG, IP, IL}. The parameters have numeric values resulted from fuzzy processes, with the exception of ND.

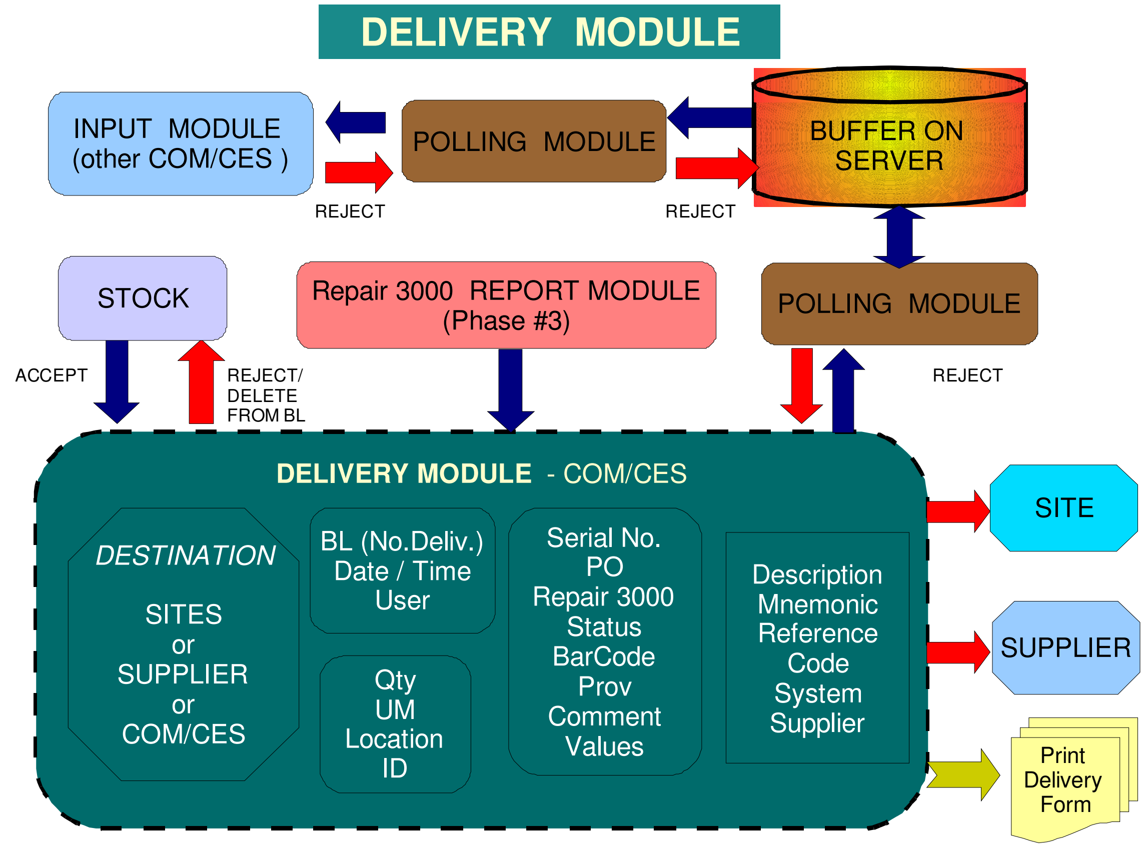

Fig.6. STOKY_RFC Delivery Module and its components

From a technical perspective, our system was developed using C++Builder 6 and its databases are administered using *PostgreSQL*, interfaced using the microolap library (Figure 9). For users, PostgreSQL is invisible and the system is runtime, the presence of C++Builder is not required. However, it is necessary to install the PostgreSQL server.

The use of STOKY_RFC logistic software improves significantly the control and management of supply activities related to ongoing company activities, as well as the repairs and other scheduled or on-demand maintenance activities. The system assists the implementation of optimal management models with regard to materials stocks, spare parts, and equipments. This software operates in real time, using a nationwide Extranet, with over 100 users.

6. Discussions

Dynamic business environments challenge enterprises in many ways. There are incentives to come up with new products for new markets and new customers. New alliances and partnerships have to be formed and new competition has to be faced. These challenges create a great strain on the existing supply chains. There is a dire need for an application technology that can make quick intelligent decisions based on real time information whenever there is such a need. Our client-server application enables the online, real time, automated management of spare parts supply and repair of technical equipments of the Romanian branch of a multinational GSM operator using an original model that combines RFID and supply chain management technologies with limited fuzzy decision making capabilities.

The system uses a polling system and a set of states for each component or machine that permits the detection of pre-defined error states of scheduled maintenance operations, together with a transition matrix that permits only certain state transformations. The system is integrated nationwide, so it can determine if and where spare parts are available, it can send requisition orders online or can generate a call for an external contractor needed to repair, maintain or resupply the system with any required part.

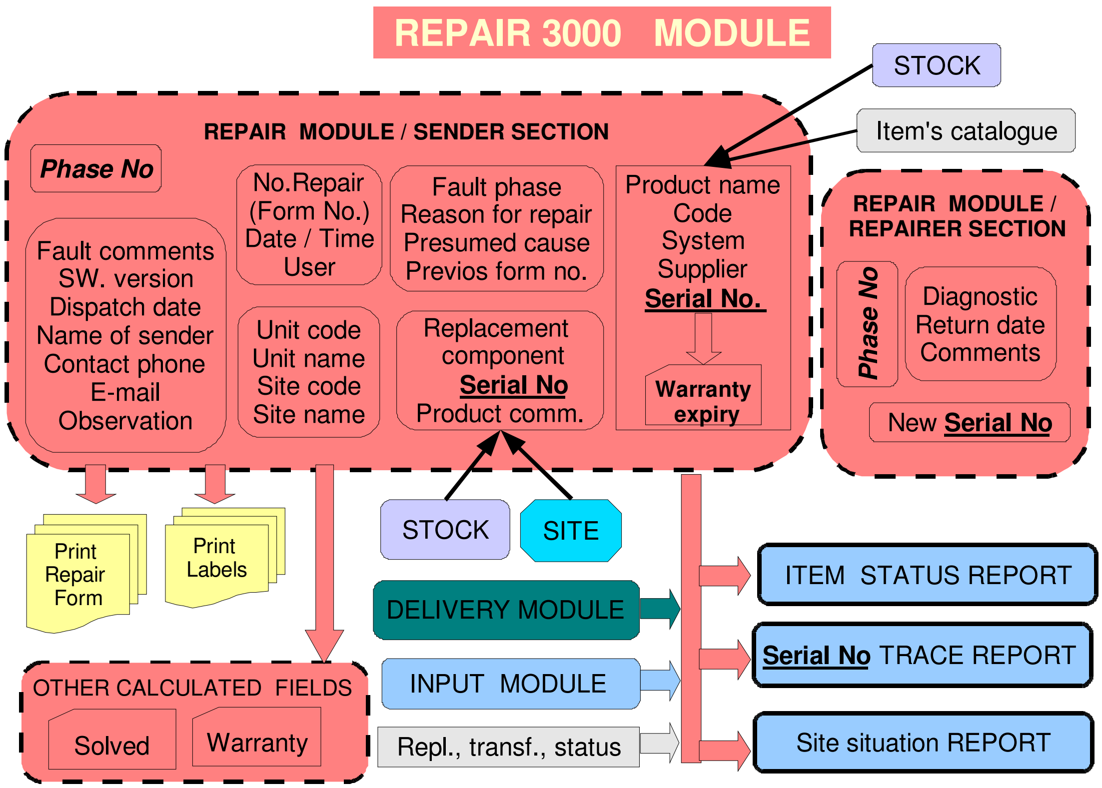

Fig.7. STOKY_RFC REPAIR3000 Module and its components

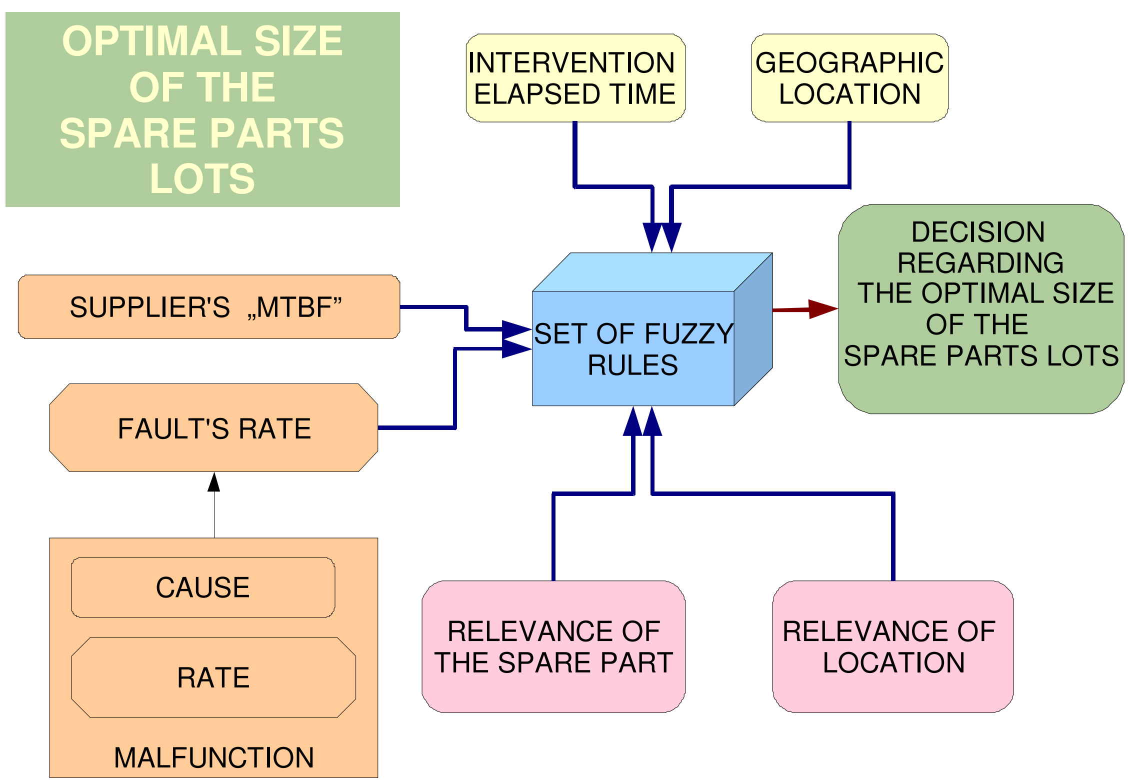

Fig.8. Identification of optimal spare parts lot size

THE APPLICATION'S STRUCTURE

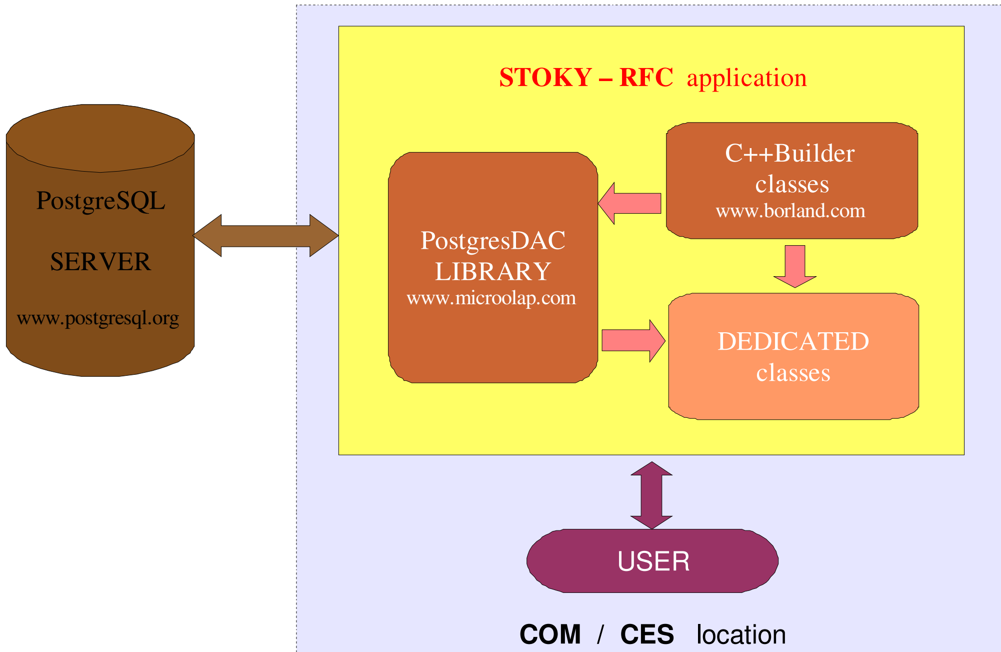

Fig.9. STORKY_RFC structure

Directions for future improvements of the system include the calibration and expansion of the fuzzy decision system, in order to optimize inventory levels and supply demands, both in terms of quantity and time, as well as the integration of the system with infrastructure development systems.

8. References

[1] Kumar, K. "Technology for supporting supply chain management: introduction", *Communications of the ACM* (44:6), June 2001, pp.58-61.

[2] van Luin, J., La Poutre, H. and Bertrand, J.W. "Learning Inventory Management Strategies for Commodity Supply Chain with Customer Satisfaction", *ICEC06*, August 14-16, Fredericton, Canada, 2006, pp.59-69.

[3] Orzan, M. "Marketing Information Systems Users: Cognitive and Behavioral Aspects", *Journal of Online Marketing* (2:1), February 2008, pp.33-43.

[4] Umeda, S. and Jones, A. "An Integration Test-Bed System for Supply Chain Management", *Proceedings of the 1998 Winter Simulation Conference*, Washington, D.C., 1998 pp.1377-1385.

[5] Huhns, M.N., Stephens, L.M., Ivezic, N., 2002, "Automating supply-chain management", *Proceedings of the first International Joint Conference on Autonomous agents and multiagent systems*, Bologna, Italy, p.1017 – 1024.

[6] Wild, T. *Best Practice in Inventory Management*, Elsevier Science Ltd, London, 2nd edition, 2002.

[7] Zangwill, W. *Nonlinear programming; a unified approach*, Prentice-Hall, Englewood Cliffs, N.J.: 1969.

[8] Cohen, M., and Lee, H. "Strategic Analysis of Integrated Production-Distribution Systems: Models and Methods", *Operations Research* (36:2), March-April 1988, pp.216-228.

[9] Sterman, J. D. "Modeling Managerial Behavior: Misperceptions of Feedback in a Dynamic Decision Making Experiment". *Management Science* (35:3), March 1989, pp.321-339.

[10] Farkas, A., T. Koltai, and Szendrovits, A. "Linear Programming Optimization of a Network for an Aluminum Plant: A Case Study". *International Journal of Production Economics* (32:2), September 1993, pp.155-168.

[11] Barlas, Y., and Aksogan, A. "Product Diversification and Quick Response Order Strategies in Supply Chain Management", *International System Dynamics Conference*, Cambridge, (MA), USA, 1996.

[12] Cakravastia, A., Diawati, L., "Development of a System Dynamic Model to Diagnose the Logistic

Chain Performance of Shipbuilding Industry In Indonesia". *International System Dynamics Conference*, Wellington, New Zealand.

[13] He, M., Rogers, A., Luo, X., Jennings, N.R. "Designing a successful trading agent for supply chain management", *Proceedings of the 5th International Joint Conference on Autonomous agents and multiagent systems (AAMAS-06)*, Hakodate, Japan, 8-12 May 2006, pp.1159 – 1166.

[14] Shen, W., and Norrie, D. "Agent-based Systems for Intelligent Manufacturing". *Knowledge and Information Systems* (1:2), June 1999, pp.129-156.

[15] Lee, H.L. and Whang, S., "Information sharing in a supply chain". *International Journal of Manufacturing Technology and Management* (20:3-4), September 2000, pp.373-387.

[16] Brown, R.G.. *Decision Rules for Stock Management*, Holt, Rinehart and Winston New York, 1969.

[17] Finkenzeller, K. *RFID handbook: radio frequency identification fundamentals and applications*, John Wiley and Sons, London, 1989.

[18] Brewer, A., Sloan, N. and Landers, T. "Intelligent Tracking in Manufacturing", *Journal of Intelligent Manufacturing* (10:3-4), September 1999, pp.245-250.

[19] Kalakota, R., Stallaert, J. and Whinston, A.B. "Worldwide Real-Time Decision Support systems for Electronic Commerce Applications" *Journal of Organizational Computing and Electronic Commerce* (6:1), February 1996, pp.11-32.