Real Time Monitoring Framework for Supply Chain Automation Using Mobile Agent Architecture

Rihards Barkans
Georgia Tech Ireland,
Athlone Business and Technology Park,
Garrycastle, Dublin Road, Athlone, Ireland

Abstract – This research examines the possibility of using the mobile agent paradigm to complement existing EPCglobal network architectures to provide real-time supply chain automation and monitoring system. The system is being implemented and tested for tracking pharmaceutical products in the Clinical Laboratory Automated Stockroom System (CLASS) project at the Galway Clinic. The proposed research evaluates the mobile agent capability to track products, collect essential data and make intelligent decisions by detecting high level events from supply chain. Currently system in use, which is based on Electronic Product Code Information Service (EPCIS) Framework by Fosstrak would be complemented by proposed multi agent paradigm and it is envisioned to set off towards the full supply chain automation whereas human involvement is minimal.

Introduction

Electronic Product Code (EPC) as a successor to barcode coding scheme is becoming a standard used in supply chain of products such as pharmaceuticals. EPC in combination with EPCIS form basis for companies to progress towards an "Internet of Things". The first stage of such networks implementation has been carried out in the CLASS project. The research presented in this article aims to investigate and quantify the benefits resulting from RFID technology and EPC system implementations.

In the supply chain of pharmaceuticals, in the first stage of development, CLASS project tracks products originating at sixteen manufacturers, continuing its lifecycle through distributors and ultimately being used up at the Galway Clinic. As there are many business steps and procedures involved for managing the product flow, high level events were observed and noted that RFID hardware is not one hundred percent accurate.

In tracking products equipped with EPC tags that missed a tracking point - these products would apparently be lost and even if they did show up later in the supply chain, then the system would not fill in where the products would have been. This results in gaps in the track and trace of miss-read products. Another limitation of the EPCglobal framework shown by the CLASS project is the difficulty in automatically collecting information whenever anomalies take place. A framework with the aim of becoming an autonomous supply chain management system must provide an appropriate "decision making" module that could manage every possible event [2].

This article intends to introduce a new approach to fulfil these requirements, exploiting the features of the Mobile Agent Architecture (MAA). MAA is a distributed computing paradigm characterized by autonomy and spread of intelligence among its elements. Mobile Agents (MAs) are autonomous programs that can be launched by a user or an application, travelling from host to host, collecting data or performing necessary tasks at each location [3]. The life cycle of an agent starts by its creation on a host that contains one or more agent servers. A predetermined itinerary is set in the newborn agent, or, the address of the next destination could be acquired through a directory service (see Fig. 1). As an agent travels through a host, it can clone or destroy itself as necessary. An agent can also share data between other agents. Agents persist until all the required objectives are achieved. An agent is an autonomous program and its lifetime may span several years. During this period many tasks may be performed, a great deal of data may be collected, and valuable intelligence may be extracted by the agents.

As stated by Lange and Oshima in [4], the benefits achieved by choosing the mobile agent paradigm can be summarized in the following seven points. They:

- reduce network load
- overcome network latency (mobile agents offer a solution to respond in real time)
- encapsulate protocols
- are autonomous and execute asynchronously
- adapt dynamically
- are naturally heterogeneous
- are robust and fault tolerant

Continuing the research conducted by Mei-Ling L. Liu in [5], this framework introduces for the first time the use of MAA within a fully functional implementation of the EPCglobal network.

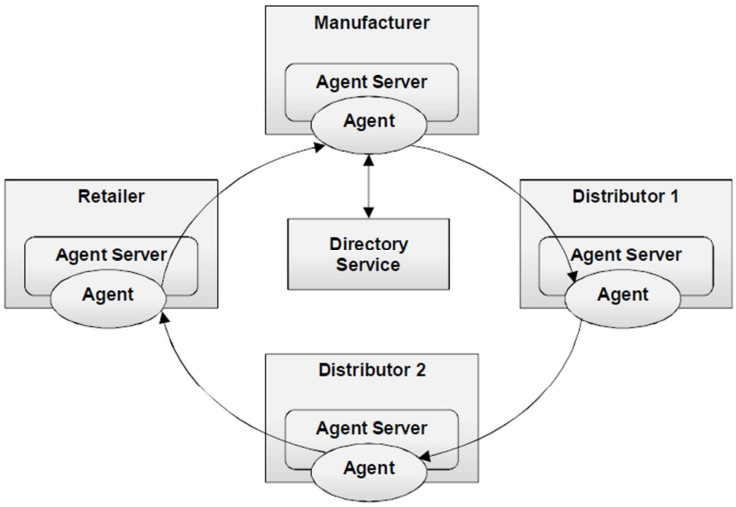

Fig. 1. Agents' mobility

Architecture Description

Introduction to CLASS Project

CLASS Project Architecture

The software architecture deployed for the project is shown in Fig. 2. The architecture of the project is based after the EPCglobal Network infrastructure [22], and is implemented on Service-Oriented Architecture (SOA) approach.

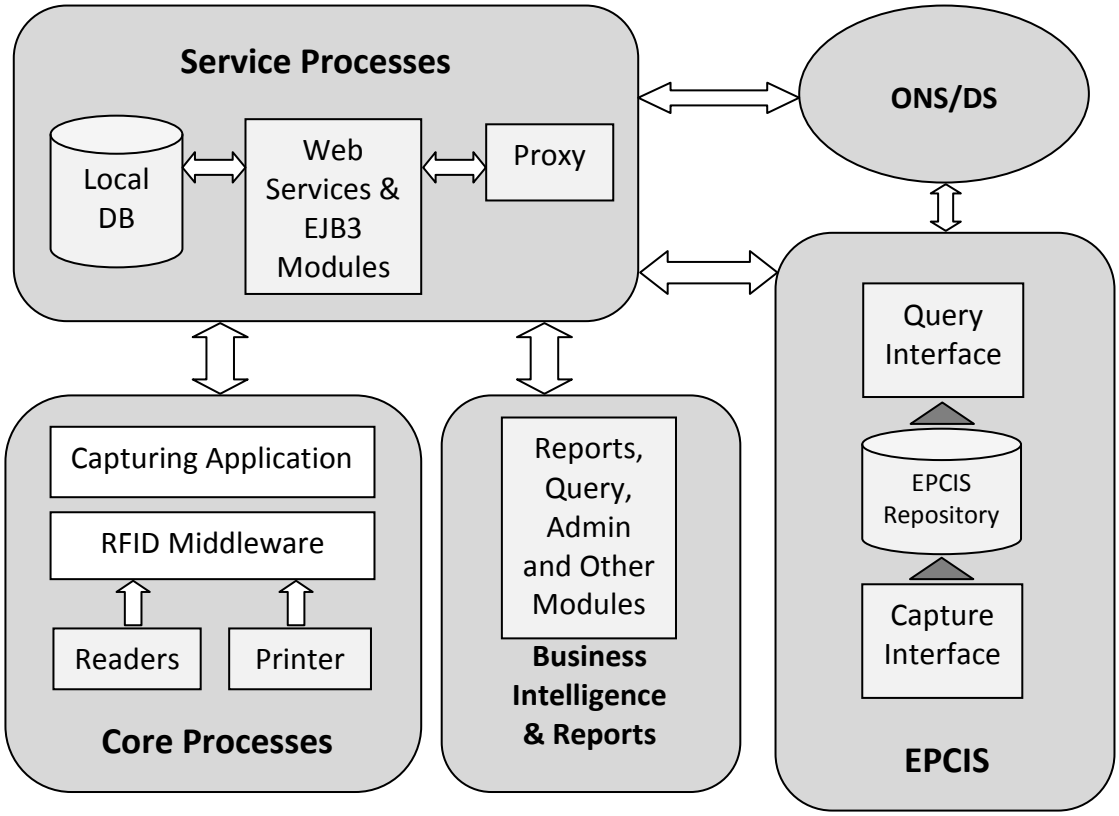

Fig. 2. CLASS – Software Architecture

The project structure is modular and can be identified into five main areas, each composed of several loosely coupled and interoperable modules:

- Core Processes (encompass the hardware devices, RFID middleware and capturing application).
- The EPCIS (standard query and capture interfaces)
- Service Processes (access point to the CLASS project services and the local DB)
- Lookup Services – Object Naming Service/Discovery Service (ONS/DS)
- Business Intelligence and Reports (represents the front-end for the operators).

Core Processes: The Core Processes provide real-time data collection from RFID hardware. This modules' function is concern with the management of tags and the related EPC information through supply chain. The raw data is processed according to EPCglobal specifications and resulting EPC events are filtered and stored in the EPCIS repository.
The lowest software layer consists of adapters, which allow communication with hardware devices such as RFID readers and printers. The adapter modules hide the details of different vendor specifications and offer a unified interface to the middleware. The middleware provides a framework through which the capturing applications can receive data read from RFID readers via Low Level Reader Protocol (LLRP) and other sensors.
The re-engineered logistic processes are embodied in the capturing applications. Capturing applications have been developed as web services client that that interface Service Processes module which runs on the application server.
Some of them are completely automated and run continuously. Others support interaction with the operators and expose web interfaces. The Capturing Applications feed the Service Processes module that forwards accordingly event data to EPCIS Repository.

The Capturing Application filters and translates the event data to comply with the standard EPCIS specifications, furthermore data is forwarded to the Service Processes module which leverages the ONS and Discovery Services to route the capture messages to one or more EPCIS repositories.
Product movement through supply chain may be characterized by the following re-engineered business processes in EPCIS as business steps *biz_step* (identified in "Ideal" Supply Chain see Fig. 4.):

- Encoding – print and write a tag. Printed RFID tags are registered with the system at manufacturing plant.
- Label – label product. Case and pallet tagging.
- Packing – aggregation of cases into pallets.
- Shipping – shipment of goods from the manufacturer to distributor and from distributor to clinic. If clinic is acting as a distributor, internal business process of product shipment between clinic branches or affiliated hospitals. Checks performed for order accuracy and aggregated products (cases and pallets).
- Receiving – receiving palette in "Goods Inward" at the distributors and hospitals. Verify order accuracy for mix and quantity of aggregated products.
- Unpacking – separation of units from the pallet.
- Repacking – repack products from different manufacturers (*biz_step* at any given potential distributor).
- Stocking – restocking products at the clinic. Monitoring product movement from the store room to the theater.
- Forward – product used at the surgery. EPC tag finishes its lifecycle.
- Quarantine – contaminated product (e.g. at the surgery product is dropped on the floor and cannot be used) is quarantined for further usage evaluation by an authoritative person.
- Holding – product is returned to the holding area (e.g. unused product is moved from the surgery to the storage area).

EPCIS: The EPCIS module is the part of the project concerning the storage of data collected from the Business Processes and the retrieval of data for the Business Intelligence modules. The EPCIS module implements EPC Network specifications stipulated by EPCglobal. It is based on the Fosstrak (former Accada) EPCIS project, an open source project certified by EPCglobal. This module consists of three components: the EPCIS Repository, whose purpose is to store the EPC events data; the underlying database is MySQL; and the Capture Interface, through which capturing applications can fill the repository with data. The Fosstrak project provides only the HTTP POST protocol binding, so a wrapper was developed to provide access as a web Service. The Query Interface, which provides data available on the repository to other applications, is implemented as a web service.

Service Processes: Service Processes module is constructed on SOA and encapsulates all the local services for the Business Intelligence and Reports, and Core Processes modules. Acquired and Filtered EPC data is received from the Core Processes module. Master data such as product name, description and batch number is added to the local database. The services from the module obtain appropriate EPCIS repository via ONS/DS to capture and query data, in combination with the master data, this module provides an interface to acquire full product information for Business Intelligence and Reports module.

ONS/DS: EPCglobal Network defines Object Naming Service (ONS) standard and currently in development Discovery Service (DS) which is not standardized yet. To lookup the

historical data of the individual products and to find associated services to the product, this implementation of ONS and DS is used. There is defined standard for the service, but individual businesses may use their own implementation on how to do lookup of the products. As products travel through the supply chain, it is important to know the historical data of the individual items, to know What, When, Where and Why?

- *What* product with EPC number, transactional data, more detailed about the products, quantity of the same class, products data – lot number and temperature of the product etc?

- *When* product arrived, left the location we can find out through the event time field?

- *Where* the product is through the business transaction, read point of the event?

- *Why* the product is – through the business step, disposition?

EPCIS infrastructure exists to answer these questions, but how to find the individual EPC's, we answer through the ONS and DS.
Service Processes module relies on the ONS and DS components of the EPC Network to locate the EPCIS servers in the supply chain. The ONS could be considered a specialized Discovery Service based on the existing DNS infrastructure. When a client queries the ONS with an EPC URN, the ONS provides the EPCIS Query Interface URL and the Discovery Services URLs' of the goods manufacturer.

Business Intelligence and Reports: the factors assessed are RFID visibility implications for the supply chain, Product flows and inventories, Traceability, and RFID Performances. The following are parts of the module:

- Product Flow - Determine at what moment a particular product was in a certain process in the supply chain. Determine the total number of packages entering and leaving the supply chain. Determine the flow (packages per day) maximum, minimum, average of packages processed during each supply chain.
- Flow Time Management - Assessing the overall and punctual flow time needed to take cases from the manufacturer to the stores (Average, Maximum, Minimum). Identify non value adding time and bottlenecks. Determines how many cases make it through the supply chain within a target flow time.
- Inventories - Assessing inventories in the supply chain (actual, average, min, max) in a selected time window. Punctually assessing inventories (for every GTIN, for every day).
- Shelf Life Management - Determine the distribution in the supply chain of cartons that have an expiration date less than or equal to the supplied date. Determine the last process in which cartons with the supplied date of expiry were seen and their remaining shelf life.
- Track and Trace - Tracking the flow of a product through the supply chain so that every place that has seen it reports to that fact. Tracking cases in the supply chain matching querying parameters.
- RFID Performance - Determine the efficiency of the devices in the supply chain. Determine the efficiency of the process, the sum of reads by RFID hardware. Compare the behaviour of processes and devices by obtaining real measures on the compatibility of each product with the RFID technologies. Determine the "lost readings" of each carton.

- Product History - Get all the traceability information for a particular product. Determine all events in which EPCIS was involved (even “non-standard” events such as readings not associated with a particular process). Identify anomalies not seen by other modules.

A key aspect of our research involves Business Intelligence module optimization in order to avoid information pulling by a daemon cron job.

CLASS Project Supply Chain

Currently in the CLASS project are identified sixteen medical product manufacturers and Galway Clinic. As for the future vision will come into play more pharmaceutical product manufacturers and hospitals as well as distribution centers and field representatives. In Fig. 7. is displayed an example of the CLASS project supply chain.

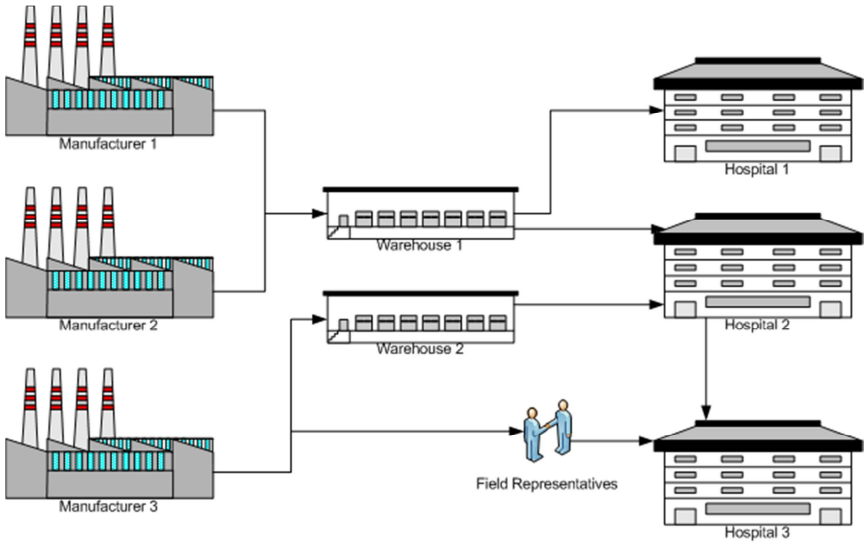

Fig. 7. CLASS – Supply Chain

Pharmaceutical products such as a clinical catheter for vascular procedures originate at a manufacturer, where the lifecycle of the product starts in the supply chain. Products make its way to the end users, generally clinics or hospitals, throughout distribution centers and specialized field representatives.

Each manufacturer stores master data about their products which contain the name of the product, description, reference number, batch number and other information. Master data is cross referenced to the EPCIS data to form the full dataset for the products.

Product Master Data

Product Name	Product Description	Product Reference	EPC	Expiry Date	Batch Number	Consignment	Company Name
Peripheral Stent	OTW .035 10mm x 59mm x 80cm	PG5910PPS	306E02B32930000000000005	2011-02-28	R0309134	No	Cordis

e.g. product EPCIS events through the supply chain. See Fig. 5. Product originates at "Manufacturer 1" (Cordis) and ultimately is being used up in "Hospital 1" (Galway Clinic).

EPCIS Product Trace

actor	event	action	biz_step	disposition	read_ point	biz_ location (GLN)	EPC	Exp Date
Cordis	object	ADD	urn:epcglobal:epcis: bizstep:encoding	urn:epcglobal:ep cis:disp:encoded	urn:epc:id:s gln:999	urn:epc:id:s gln: 539701907	urn:epc:id:s gtin: 539701907. 0000.5	2011-02-28
Galway Clinic	object	OBSERVE	urn:epcglobal:epcis: bizstep:receiving	urn:epcglobal:ep cis:disp:available	urn:epc:id:s gln:098	urn:epc:id:s gln: 539701999	urn:epc:id:s gtin: 539701907. 0000.5	2011-02-28
Galway Clinic	object	DELETE	urn:epcglobal:epcis: bizstep:forward	urn:epcglobal:ep cis:disp:sold	urn:epc:id:s gln:097	urn:epc:id:s gln: 539701999	urn:epc:id:s gtin: 539701907. 0000.5	2011-02-28

Rationale for Multiagents in CLASS Project

The CLASS Business Intelligence module requests data from EPCIS repositories. Vast amounts of raw data are generally associated with RFID solutions. This historical data can be accessed via Fosstrak EPCIS query interface. Queries are executed manually by a user or on a weekly basis through cron daemon where a list of jobs and instruction are kept, which means that the business intelligence is acquired manually based on historical data.

A more dynamic and active approach is offered by multiagents. Multiagents collect data and search for high level events in real time. Transparent to the users of the system, multiagents are created in response to EPCIS events and dispatched through the network. While multiagents function in the background, they offer active information management and are able to perform tasks that lead to supply chain process automation. Mobile agents are truly distributed and offer equity to all supply chain partners as opposed to a client/server architecture.

It is very beneficial for the project that both systems, multiagents and the cron daemon queries, may work in parallel within EPCglobal architecture. While business processes are actively managed and valuable information reported by multiagents in real time, the users may use the existing EPCIS query interface as a passive way of extracting intelligence about the supply chain. Both systems may benefit from each other by cross-validating the obtained data. RFID hardware is not one hundred percent accurate and precision is very important, especially in healthcare. This requires repeated scans and additional tasks by human operators. In many situations, containers with goods automatically pass through RFID scanning gate and in some cases there is a need for a human operator to perform RFID scans manually. The ideal situation is to remove human intervention from the supply chain and automate the process through an intelligent supply chain. For this purpose we need to introduce intelligent algorithms that could be embodied in multiagents to automate the processes. Task data can be embedded into an agent, after dispatching it into the network the agent becomes an autonomous program [23]. By adapting dynamically, an agent can learn the network topology and hardware performance, this includes specific RFID reader accuracy, and at later stage it can notify other agents through inter-agent communication about irregularities encountered which would help other agents to analyze the situation and to make intelligent decisions.

Incorporating Multiagents in EPCglobal Architecture

The EPCIS standard provides the foundation necessary for the capture, communication and dissemination of EPC data about goods and services products [24]. EPCIS is modular and has a set of interfaces for capturing and sharing information. By employing those interfaces, we can easily implement a multiagent framework that complements EPCIS and provides a mechanism to manage the data in real time between trusted business partners and the other parties that wish to integrate their Supply Chain Management (SCM) and Enterprise Resource Planning (ERP) systems through the EPCglobal architecture.

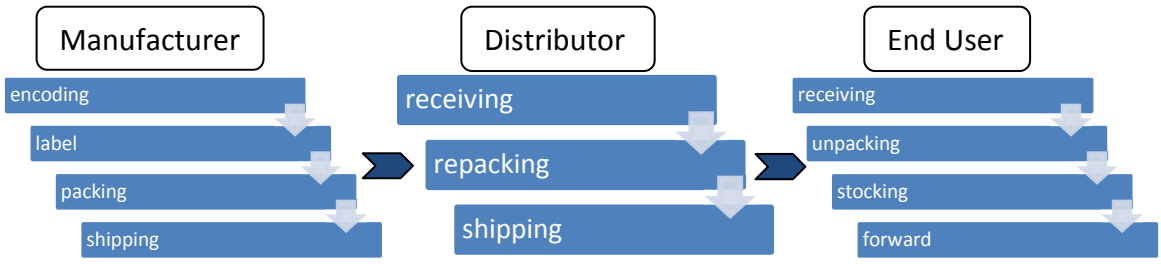

Fig. 4. EPCIS Business steps

Very important business information [25] like time, location, disposition and business steps that are associated with individual items in supply chain can be shared via EPCIS standard. Inspection of each EPCIS business step *biz step* field suggests that it is feasible to segment supply chain as each business step may be detected by physical RFID reads. Items move through supply chain according to business

steps. By segmenting the supply, we can achieve a greater degree of control and visibility over the processes. Therefore, it is feasible to manage each business step independently. Business steps should be linked together with the other business steps for the information to be exchanged and processed. In this scheme that matches the CLASS project, product movement follow the EPCIS Business steps shown in Fig. 4. Our software creates agents that travel alongside the products through supply chain. Agents need agent servers in order to travel from host to host. We situated mobile agent servers at the physical location of each business step. When a new item is created at manufacturer, it travels through distribution to a clinic until the item is sold (an item travelling downstream). In the same way, a new "downstream" agent is created for a set of products. It would be computationally expensive to create a new agent for an individual item; therefore an agent is created for a pallet with multiple items. Further agent will split into multiple agents when the pallet is segregated. The function of downstream agent is to travel through hosts and collect EPCIS event data. Conversely, if an item is refunded or returned to the previous business step, then we can say that the item travels upstream. This requires "upstream" agents.

Architecture of the Multiagent Application

The application was developed in Java. Multiagent paradigm was implemented with Java Remote Message Invocation (RMI) alternatively any relevant multiagent framework can be modified to suit the project needs. The Fosstrak project was modified to create a simulator for EPCIS events. The simulator uses real data acquired earlier from the manufacture, distributor and the Galway Clinic to simulate supply chain managed by multiagents as the first step before implementing it into the real life. The simulator reconstructed EPCIS events that actually occurred in the CLASS project in the past. By reconstructing real scenarios, we could study the concept of multiagent usability and performance in supply chain.

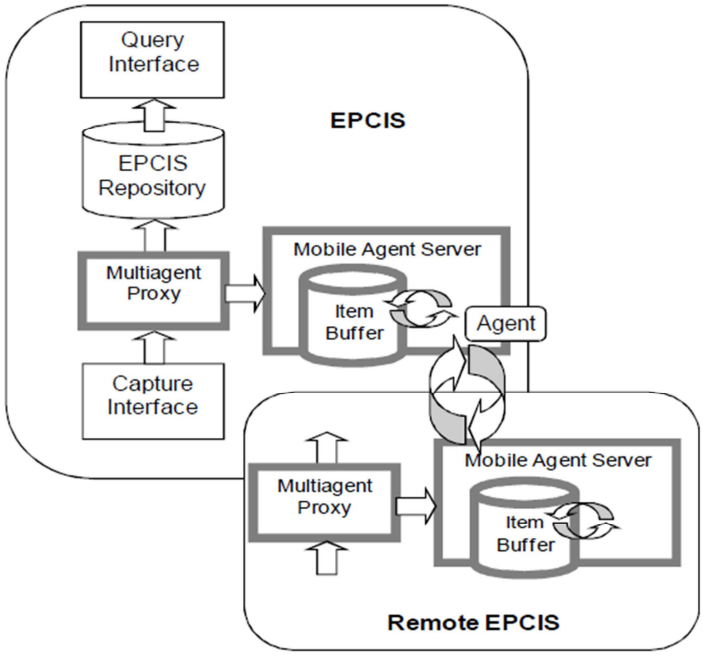

Fig. 5. Multiagent Application Architecture

1) *Mobile Agent Server:* The main component of the application is the agent server. As explained earlier, the supply chain is segmented into logical locations, according to business steps. Each segment has its multiagent server which is responsible for creating, cloning, sending and receiving mobile agents.
2) *Mobile Agents:* There are multiple types of multiagents. The main type is a downstream agent which goes along with the products through supply chain and carries the EPCIS event data

generated at each business step. When products arrive to the next business step, an upstream agent is generated and travels to the previous business step notifying that the products have been received. There are two other types of agents used for searching and reporting purposes.

3) *Item Buffer:* As shown in Fig. 5, at each business step there is a multiagent buffer. The Item Buffer is a temporal storage of EPCIS events that are generated onsite, it also has a timeout. Dealing with groceries, items tend to stay at each location for no longer than two to three days. For this purpose the item buffer also may be programmed to check the expiration date of the products and make a report in case there are expired items. If the buffer timeouts a specific product which has stayed for maximum allowed period of time, it means that the item has been lost at the specific location (business step) or for any other reasons the items has been last seen at that location, but no more, because it never received notification agent from the next business step and was not removed from the buffer. This is because the item has not been received and the notification has never been sent to the previous business process, thus the item staying in the buffer.
4) *Multiagent Proxy:* Multiagent Proxy forwards generated events from the Capture Interface to the Mobile Agent Server. Mobile Agent Server contains multiple logical servers, one logical server per business step.
5) *Agent Creation:* Downstream product movement is from the manufacturer to the distribution centres and retailers. It is a great advantage when agents are moving along with the items. This parallel movement of agent and product allows the full history of the product to be kept as it moves through the supply chain. Collected information provides the business partners with the historical data of the product upon delivery or upon request. Each time a set of items is produced and aggregated into a pallet then EPCIS events are generated followed by the creation of a new downstream agent for a pallet. Multiagent logic is based on creating and sending a notification agent upstream when an EPCIS event is generated. Once the notification is received by the previous business process and satisfies the conditions (see Fig. 6), downstream agent may move down the supply chain to the next business process. If the condition is not satisfied or partially satisfied, notification is forwarded further upstream. Upstream agent movement from the retailer to the distributor or distributor to the manufacturer is used for delivery confirmation and invoicing as well as for damaged products and refunded products.

Benefits

The use of this Multiagent Architecture has many benefits for supply chain management. Some of these benefits are as follows.

- *Full History of the Product* – Mobile agents traveling alongside the products collect considerable amounts of data, from which the users can extract necessary information.
- *Finding Missed RFID Reads* - Mobile agents are able to intelligently differentiate via inter-agent communication and historically accumulated data about high level events that occur in the supply chain such as missed reads by RFID hardware.
- *Track and Trace* – Multiagents may be released in supply chain by querying significant locations about the specific product. In this scenario multiagents would act similar to search bots on World Wide Web.
- *Order Confirmation and Automatic Invoicing* – Multiagents acting as a delivery confirmation, sending and receiving parties can be invoiced accordingly. Also if the agent can distinguish that in fact the product is missing and it is not just a missed RFID read, then product can be reordered automatically.
- *Reduced Human Interaction* – Since the Multiagent Architecture can detect missed reads it eliminates the need for staff to check the accuracy and quantity of the pallet mix. The Multiagents also inform about products that have expired and need to be taken out of supply chain or the inventory of a hospital

- *Other Benefits* - If a product is close to expiration date, adequate notification can trigger an event that would cause the product to be shipped to the location where it is low in stock or ready to use on arrival. The term “on consignment” was introduced in the CLASS project to deal with the high price medical products that are close to expiration date, to be shipped between hospitals, to eliminate high losses by the manufacturers.