UNIVERSITY OF NIGERIA, NSUKKA

FACULTY OF ENIGENEERING

DEPARTMENT OF ELECTRONIC ENIGINEERING

A PROPOSAL WRITING IN PARTIAL FULFILMENT OFNTHE REQUIREMENT IN TECHNICAL REPORT WRITING

(ECE 494)

TOPIC

DESIGN AND IMPLEMENTATION OF GOODS TRACKING DEVICE ON FREIGHT AND SHIPPING SERVICES

BY

NGWU URUKANACHIM JEREMIAH

2017/242561

LECTURER

PROF. O.N ILOANUSI

SEPTEMBER, 2021

TITLE

Design And Implementation Of Goods Tracking Device On Freight And Shipping Services

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The purpose of this project is to find a solution to the reoccurring problems challenging the freight and shipping services using a GOODS TRACKING DEVICES Monitoring sensor data in container is a big issue in logistics system. Containers in logistics system generate huge amount of sensing information which requires handling efficiently to provide response to the user queries. Moreover, collection of sensor data from sensor tag by using heterogeneous reader devices is also a big challenge in sensor tag deployment. To achieve these goals, a flexible middleware is needed that provides glue between applications and the heterogeneity of devices by facilitating optimized set of services at high performance to a scalable number of users. In this paper, we propose Sensor Tag Middleware architecture that incorporates characteristics of standard EPCglobal middleware with several unique features such as reader abstraction, integration, high performance and scalability. In order to illustrate our proposed middleware, we implement it for monitoring containers in port logistics.

INTRODUCTION

Sensor technology is now being used more frequently in container management (CM) in port logistics. By bridging the gap between the virtual world of IT systems and the real world of products and logistical units, sensor technology has the potential to solve many existing business difficulties in both sectors. Safe container transportation, more efficient material handling processes, quality control of temperature sensitive perishables with continuous stationary monitoring, elimination of container damage due to huge inside pressure, and automatic tracking of product location in supply chain are just a few of the many advantages.[1]

This is a time-consuming and laborious operation that has been handled by abstracting a system's physical perspective into a logical model and implementing that logical model as middleware. The fundamental goal of middleware systems is to offer powerful abstractions that codify the domain's essential requirements and notions while also allowing it to be extended to the physical environment. Furthermore, a well-designed middleware architecture allows for application modularity, adaptivity, and repairability by allowing for architecture flexibility as requirements change.

We developed a cutting-edge sensor tag Middleware for Monitoring Containers (MMC), that combines sensor tag and RFID (Radio Frequency Identification) middleware and is compatible with the EPCglobal architecture.[1] We studied the requirements that sensor tag middleware components should meet in order to manage containers with massive deployments of tags with sensing capabilities, diverse heterogeneity in reader devices, and the amount of data that readers acquire in this paper. This paper's key contribution is the creation of software components for sensor tag middleware that address both application needs and sensor technology constraints.

BACKGROUND OF STUDY

In general, a tracking system is used to observe moving people or things as well as provide a model with a timely ordered sequence of their location data. Importation tracking requires the use of information technology. The term "information system" refers to a system that involves people, data records, and activities that handle data and information in a company, as well as manual and automated operations. In a strict sense, an information system is application software that is used to store data records in a computer system and automates some of the organization's information processing activities. Corporations are also using location-tracking technology to streamline supply chains in order to get products to market faster, as well as to track assets and prevent unnecessary inventory loss.[2]

The fastest way to find out where your shipment is is to use NIPOST's online tracking. There's no need to call customer care when we can provide you with real-time updates on the status of your package as it travels through the NIPOST Network on its way to its final destination. Companies will soon be able to track your location as well. Imagine walking through your local grocery and receiving an electronic coupon for your favorite cereal as you pass through the aisle. Increased demands for efficiency have led to information technology being recognized as a significant and innovative addition to goods monitoring systems.[3]

Computer and telecommunications technology have advanced at a breakneck pace in the recent decade. With increased computer power, innovations in data transmission, and creative and user-friendly graphic interfaces, shipping ports now have unparalleled capacity to gather, store, analyze, and exchange information with stakeholders both inside and outside of ports.[2]

STATEMENT OF PROBLEM

The current growth rate of clients calling tracking agents and visiting the NIPOST office to ensure the status of the parcels they shipped through the NIPOST service channel is far too high. Owing to insufficient methods of communication with the company while on the road, the NIPOST agent may not be able to say whether the parcel has been delivered or not, and it is also more prone to error due to data loss.

The old system's data entry and validation are done manually, and the system uses the data entered as the basis for the tutorial schedule, while all papers delivered to clients are derived from the data collected and the result of the data processing. This procedure is strongly associated to extensive manual processing, limited tracking visibility, and poor customer service (at least perceived). The importance of an effective chain-of-custody proof-of-delivery (POD) solution is developing in day-to-day operations.

Because they are market and prone to error, the amount of duty payable must be determined to the penny, and goods are recorded and the amount of duty payable must be calculated to the penny, since they are market and prone to error and fractions of a penny being disregarded. Customers would like that their goods clear on time due to their eagerness, however because there are so many actions involved, there is a delay in processing, and items may be mixed up during numbering and correctly assigned responsibilities for identical items. The value of storing information such as and other vital documents might be high, leading to the client's distrust in the agency.

AIM AND OBJECTIVE OF THE STUDY

In view of the phenomenal growth in transportation industry and the trend of information technology in delivery first class service to the customer with minimal cost. This project aims to develop online and real-time transit parcel monitoring system to the NIPOST clients. Following objectives:

1. To provide fastest and most reliable service to the client at all time, all days through mobile technology
2. To prove self-monitoring environment for the client’s parcel or freight on transit.
3. To ensure proper processing and dissemination of key shipping information for quick vessel turnaround.
4. To provide self-management and the unbending security in monitoring client’s parcels on transit
5. To ensure efficient management of port operations, optimal allocation and use of resources.

SIGNIFICANCE OF STUDY

The advantages of an E-tracking system over the existing system, i.e. the manual way of registering and monitoring the movement of freight or any parcel, will be used to implement the study's objectives. Users will be able to track and access resources digitally instead of hauling books, journals, periodicals, and other publications. As a result, the paper documents may be subjected to rapid wear and tear. Additionally, it will allow users to view the current location of their items or parcels. The proposed system will ensure security by enabling each ship and parcel carrier in transit to be tracked. The solution will also eliminate some of the tedious manual labor that currently exists in the manual tracking system.[3]

In a variety of ways, the software developed will enhance port importation management:

1. Keep a centralized database for all of your data.

2. Using serial query language, easily retrieve past client information from the system.

3. For security reasons, keep a keen watch on the client's products and information.

4. Be aware of the number of things they obtain on a daily basis.[2]

SCOPE OF STUDY

The goal of this research is to figure out how to computerize the import duty processing system. It is solely applicable to the operation of a shipping company's system.

METHODOLOGY

Architecture of MMC

Figure 1 shows the detailed MMC architecture. It consists of three layers: Application Layer (Web Service), Core Engine Layer (MMC Engine- Sensor Middleware) and Reader Abstraction Layer (Adapter Framework).

1) Application Layer: This is the Middleware's upper layer, which allows customers the access to sending data via a Web Service API. Using the Common Application Interface, MMC provides application independence. We used the ALE Tag Memory (TM) Spec. API, Reading API, Writing API, Logical Reader API, and Access Control API, which are all offered by EPCglobal. To transmit queries to the MMC sensor Tag middleware, web applications employ conventional protocols (HTTP, TCP, and File).

2) Core Engine Layer: The Core MMC Engine is part of this layer. It's a sensor middleware that includes modules, a database (SensorDB), and loggers. SensorDB holds all sensing data, while the Indexing Module uses the CQ Index to provide quick responses to user requests. The Query Processor Module parses and processes the queries that applications feed to it. The logical name of the tag memory address seeking sensing data is contained in the query. The Sensor Tag Address Mapping Module (SenTAM) converts the logical name for sensing data given by the user into a physical address value. The Event Collection Module collects continuous sensor events from the reader. The Event Filter Module filters streaming sensor events based on user-defined filtering conditions. Finally, the Report Generator Module generates sensor reports, and middleware transmits the generated report to the associated user in response to the user's query. Furthermore, MMC consists of three types of Loggers (TCP Logger, HTTP Logger, and File Logger) for logging users who contact the middleware asynchronously.

3) Reader Abstraction Layer: Reader Abstraction Framework is the lowest level of MMC. It is used to give the middleware reader freedom. Reader protocol and Adapters are the two sorts of modules that make up this system. In one hand, the Reader protocol module is utilized to communicate and transfer data handling between EPCglobal RP compatible readers and the middleware. Individual adapters are used for communication between the reader and the middleware for each non-RP compliant reader. As a result, Reader Abstraction Framework creates

a seamless connection between MMC Sensor Tag Middleware and various types of sensing data readers. Users can configure, monitor, and control reader devices using this layer without knowing anything about reader kinds. To control reader-specific features, it supports an XML-based reader control mechanism.

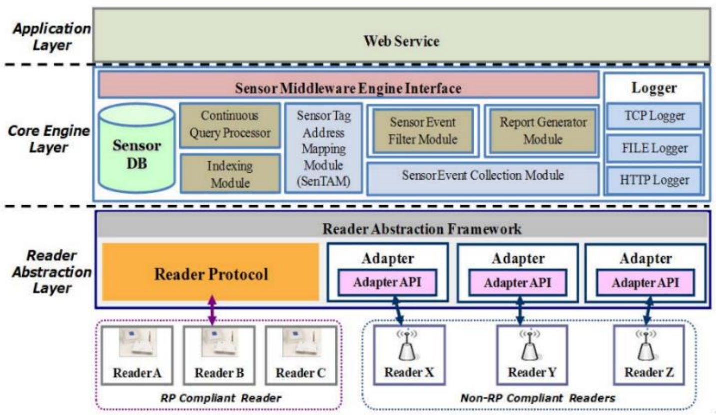

Figure 1 : Reader abstraction framework for MMC

Implementation of MMC

MMC sensor tag middleware has been implemented for container monitoring in port logistics. Containers must be monitored closely in port logistics systems to ensure their safety and security. Sensor tags are connected to each container in this case to identify and track it. Each tag has its own user memory, which is used to store sensing data from the container to which it is attached. We analyze four implementation scenarios to illustrate our MMC implementation:

(1) Tag Lists Collection

(2) Sensing Information Collection

(3) Door Status (Open/Close)

(4) Alarm Notification

Figure 2 depicts all four types of implementation scenarios, along with a thorough discussion of each. 1) Tag List Collection: Each container has a sensor tag attached to it. The reader gets the tagID of each container and provides it to the middleware based on the clients' requests. The middleware gathers all of the tags and generates a list of monitored containers in the yard. The list of tags obtained by MMC is shown in (1) in Figure 2. 2) Gathering of Sensing Data: (2) in Figure 2 depicts my MMC's collection of sensing data. With each container, we deploy a number of sensor devices. A heat lamp, for example, is mounted to the container and produces both heat and light. The temperature sensor gathers data and transmits it to the sensor tag affixed to the container. The Thermo Threshold meter keeps a record of the temperature.

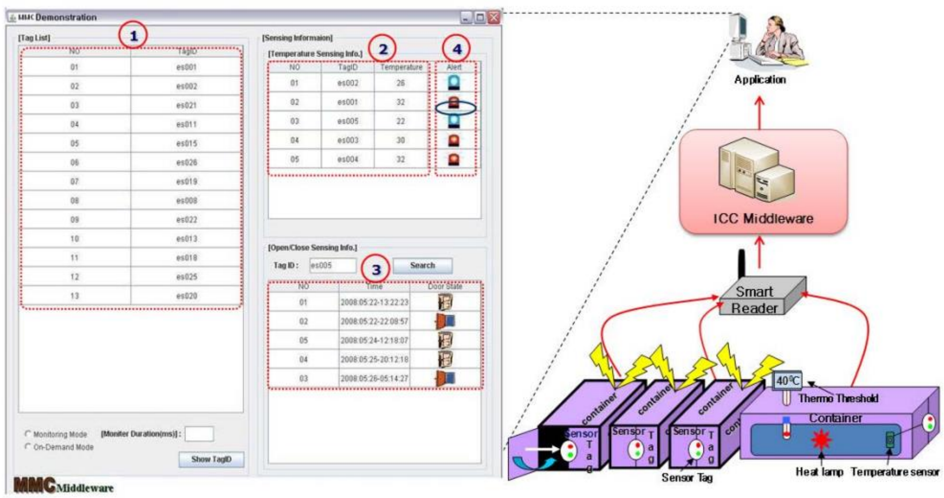

Figure 2 : GUI of MMC

Whether or not the temperature rises above the set point. As a result, the temperature readings obtained are saved in the tag memory. When a user requests any sensing data, the Reader reads the tag's memory location to retrieve the data and transmits the sensing value to the MMC sensor tag middleware. Status of the door (open/closed): One of the most important requirements for ensuring container security in logistics systems is to monitor door status. We used a light sensitive sensor within the container to meet these requirements, which monitors light intensity in the container and determines whether the door is open or closed. (3) in Figure 2 depicts the condition of each container's door in our logistics setup. 3) Alert Notice: If an unpleasant scenario arises in the container, such as overheating, overpressure, or an opened door, application users must send an alarm notification. Users can set alarm conditions with MMC. When the recorded sensing value exceeds the threshold limit, the middleware sends the relevant user an asynchronous alarm notification. Figure 2 shows an automatic alarm warning issued by MMC when the container's inner temperature exceeds 300 degrees Celsius. As a result, the user receives reliable system services.

D. MMC Performance Evaluation 1) Experimental Environment: I choose Java as my programming language and the JDK 5.0 library as my library. The experimental platform is Windows, with Windows XP as the operating system, 1GB of main memory, and a Pentium IV 2.6GHz processor. 2) Experimental Results: I compared the cost of construction with naïve sequential matching, which is a query indexing technique that does not use a query indexing scheme. The cost of continuous query processing increased significantly as the number of continuous inquiries increased in sequential matching processing. When 1000 queries are registered in sequential matching, the CPU use is 100%. As a result, no further processing is feasible. However, only 10% of the CPU is used in Query Indexing. Figure 3 shows how the CPU occupancy rate fluctuates as the number of continuous inquiries increases.

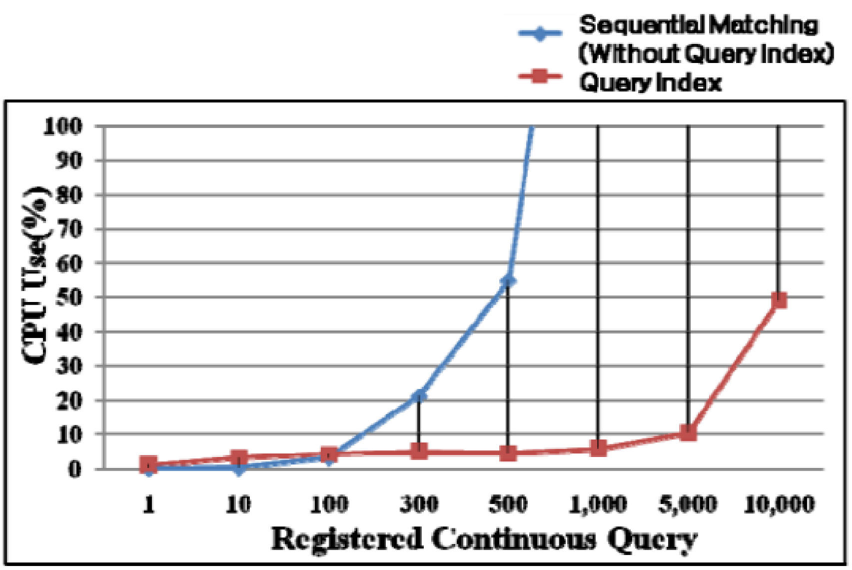

Figure 3 : CPU cost

GANTT CHART

DAYS/ TASKS	1	2	3	4	5	6	7	8	9	10
TASK 1										
TASK 2										
TASK 3										
TASK 4										
TASK 5										
TASK 6										
TASK 7										
TASK 8										
TASK 9										
TASK 10										

Table 1 : GANTT CHART

Task 1 – comparative analysis on available topics

Task 2 – choosing of personal topic

Task 3 – research on proposed project topic

Task 4 – consultation of books related to my project topic

Task 5 – researching of components related to my topic

Task 6 – implementation of my project topic

Task 7 – initial testing of my work

Task 8 – correction of observed errors

Task 9 -- initial testing of my work

Track 10 – presentation

BUDGET

OPERATING EXPENSES	ESTIMATED N
Advertising	100,000
Staff Recruitment (6 employees)	200,000
Purchase of Material	1,000,000
Transport	150, 000
Food	250,000
Data Subscription	200,000
Rent	550,000
Health Insurance	1,500,000
Maintenance and Repairs	450,000
Total Operating Expenses	4,400,000

REFERENCE

[1] G. Kim, M. K. Uddin, and B. Hong, “Design and implementation of sensor tag middleware for monitoring containers in logistics systems,” *Proc. - 2009 3rd Int. Conf. Sens. Technol. Appl. SENSORCOMM 2009*, pp. 393–398, 2009, doi: 10.1109/SENSORCOMM.2009.65.

[2] Ikuejawa Abiola Oladeji, “DESIGN AND IMPLEMENTATION OF IMPORTATION TRACKING SYSTEM – Project Gist International.” https://projectgist.com.ng/2017/02/12/design-and-implementation-of-importation-tracking-system/ (accessed Oct. 16, 2021).

[3] iproject, “DESIGN AND IMPLEMENTATION OF AN ONLINE REAL-TIME PARCEL TRACKING AND M...” https://iproject.com.ng/computer-science/design-and-implementation-of-an-online-real-time-parcel-tracking-and-monitoring-system-a-case-study-of-nigerian-postal-service-nipost/index.html (accessed Oct. 16, 2021).

[4] A. Kabir, B. Hong, W. Ryu and S. Ahn, ”LIT Middleware: Design and Implementation of RFID Middleware”, In proceedings of Dynamics In Logistics, LDIC 2007, Bremen, Germany, August 2007.

[5] C. Floerkemeier, M. Lampe “RFID middleware design – addressing application requirements and RFID constraints” , Joint sOc-EUSAI conference, Grenoble, October 2005.

[6] T. Liu, M. Martonosi, “Impala: A Middleware System for Managing Automatic Parallel Sensor Systems” PPoPP’03, June 11-3, 2003, San Diego, California, USA.

[7] K. Aberer, M. Hauswirth, A. Salehi, “A Middleware for Fast and Flexible Sensor Network Deployment”, VLDB’06, September 12-15, 2006, Seoul , Korea.

[8] S. Madden, M. Shah, J. M. Hellerstein, V. Raman,"Continuously Adaptive Continuous Queries over Streams", in Proc. of the 2002 ACM SIGMOD Conf., on Management of data, Wisconsin

[9] A.Arasu, S.Babu, J.Widom, "The CQL Continuous Query Language: Semantic foundations and query execution", The VLDB Journal vol.15 Number 2, June 2006. [10] Wu K. L, Chen

S. K, Yu P S. “Processing continual range queries over moving objects using VCR-based query indexes”, Proceeding of Internat onal Conference of Mobile Ubiquitous Systems, 2004.

[10] EPCglobal Inc., “The Application Level Events (ALE) Standard Specification Version 1.1” February 27, 2008.

[11] EPCglobal, “EPCTM Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communication at 806 MHz – 960 MHz Version 1.0.9”, EPCglobal Standard, January 2006.

[12] EPCglobal Inc., “The Reader Protocol Standard, Version 1.1”, Ratified Standard June 21, 2006.

[13] EPCglobal Inc., “The EPCglobal Architecture Framework, EPC global Final Version 1.2”, Approved 10 September 2007.