A MULTI-AGENT BASED SYSTEM RFID MIDDLEWARE FOR DATA AND DEVICE MANAGEMENT

LIBE V. MASSAWE, FARHAD AGHDASI AND JOHNSON KINYUA

ABSTRACT

Radio-frequency Identification (RFID) technology promises to revolutionize business processes. While RFID technology is improving rapidly, a reliable deployment of this technology is still a significant challenge impeding its widespread adoption. In this paper we provide a brief overview of some common fundamental characteristics of RFID data and devices, which pose significant challenges in the design of RFID middleware systems. In addition, the development of a multi-agent RFID middleware solution to address the RFID data and device management challenges is discussed.

Key words: RFID, RFID middleware, data management, multi-agent systems

1. INTRODUCTION

Radio frequency identification (RFID) is a technology that allows an object, a place or a person to be automatically identified without the need for physical or visual contact. With potentially significant applications such as smart homes and social devices [1], and the continuous decrease in both size and the cost of the passive RFID tags, it is predictable that every object could be tagged in the near future. The extensive use of an electronic tagging technology at the item-level will generate a flow of product information that is several orders of magnitude greater than it is now. For example, industrial deployments in distribution centers and stores will have hundreds of readers deployed across multiple locations. Handling such large volumes of real-time RFID data streams resulting from multiple readers and tags, intelligently transforming them into useful information, and concisely interacting with other enterprise software applications pose some interesting challenges for both data and device management.

The hardware technology for RFID, that is readers and tags, is becoming affordable resulting in most organizations showing interest in deploying the technology. However, one of the major challenges organizations are now facing is integrating an RFID system with the rest of the enterprise applications. According to Forrest research [2], a new breed of software called RFID middleware is required in order for the organizations to take full advantage of long-term benefits offered by RFID implementations. RFID middleware is the component that manages RFID data by routing data between tag readers and the multitude of enterprise information systems. Many early RFID middleware solutions focus on features like reader integration and coordination and basic data filtering capabilities. As the market matures, middleware platforms will need to include a deeper set of capabilities including: reader and device management, data management, application integration, partner integration, process management and application development, packaged RFID content, and architecture scalability and administration [2].

In this paper we discuss some of the common RFID data and devices characteristics which pose many interesting challenges in the design and development of the RFID middleware platforms. We also present our proposed RFID middleware framework based on intelligent software agents. Our intension is to take advantage of powerful tools and capabilities offered by software agents' technology [3], to model and implement the complex middleware functionalities.

2. CHARACTERISTICS OF RFID DATA AND DEVICES

2.1 RFID Data

Despite the diversity of RFID applications, RFID data share some common fundamental characteristics, which pose great challenges and they need to be fully considered in the design of RFID middleware systems.

Streaming and large volume
RFID data are generated quickly and automatically, and accumulated for tracking and monitoring. The data generated can be enormous, which requires a scalable storage scheme, which facilitates efficient query execution and fast updates. In addition, due to the nature of RFID applications which demand track and trace queries for individual items and the large volume of data, the lineage tracking problem is more critical and challenging in RFID than in traditional data warehousing [4].

Implicit semantic data
Raw data generated from an RFID system can be seen as stream of tuples comprised of the reader EPC (Electronic Product Code), the observed EPC value of an RFID tag, and the timestamp when the observation occurred [5]. This data carries implicit information about business processes such as changes of states, change of locations, and containment relationship among objects. For example, the detection of number of tags at the dock door over a certain period of time should be automatically translated as “shipment arrived” event. Extracting this implicit information from these raw data is the most interesting and challenging issue in an RFID data management system. Thus, a framework is needed in order to automatically transform the simple observed data into business logic data which enterprise applications such as inventory tracking and resource planning can use.

Inaccuracy data

The location tolerance that makes RFID tags easy to read also makes it difficult to understand whether a tag is in fact in the reader's prescribed zone, or whether the read tag is simply passing by [6], “false positive reading”. Missed reads are also an unfortunate reality with RFID systems [7]. While reader performance is improving, cost pressures will always dictate that some RFID systems be used at the limited performance. In addition, problems such as reader interference and multipath fading will also cause many reads to be missed, “false negative reading”. Such erroneous and unreliable data must be semantically filtered online before it is transformed into business logic data.

Data Redundancy

RFID data on average is less useful than other data streams [1]. For example, in traffic monitoring and financial applications, every record might be useful for further analysis. On the other hand, in the case of RFID data, we should be able to identify data that has been read multiple times. The less useful part of RFID data is the data that are continuously reported after the initial reading. Data redundancy can also be caused by an item being in vicinity of more than one reader [8] and as a result its data is read by more than one reader. Therefore, it is necessary to have a filtering mechanism to reduce RFID data redundancy before processing the observed raw data.

Spatial and Temporal

In general, most of RFID based applications are not interested in individual readings in time or individual devices in space, but rather in an application-level concept of *temporal* and *spatial granules* [9]. These granules define the lowest-level, atomic unit of both time and space in which an application is interested [9]. Therefore, the RFID data management system needs to have explicit temporal and spatial data models for RFID data to support tracking, tracing and monitoring application queries.

RFID data dissemination and Integration

The raw data generated from RFID network itself is not valuable unless it is correlated to other information. For example, when the dock door RFID reader registers the arrival of a new pallet, it is vital to be able to correlate this arrival event of a pallet and all uniquely tagged cases to a purchase order, an invoice or advanced shipment notice. This means that, RFID data have to be integrated with existing legacy enterprise applications. In addition to that, the information captured by a reader is usually of interest to a diverse set of applications across an organization and its business partners. The captured RFID data must thus be broadcast to the entities that indicated an interest in the data. Due to the event-driven nature of many processes observed with the help of RFID systems, there is a need to support asynchronous messaging as well as a query-response model [10, 11].

Legacy applications that are not designed to handle streaming data might need to receive batched updates on a daily schedule [10]. This requires an RFID data management system to be easily configured into different applications.

2.2 RFID Devices

In most practical RFID implementations, the system is made up of multiple readers; also the readers must interact with other devices such as sensors, PLCs (Programmable Logic Controllers), and human interfaces. Thus, device management is necessary for monitoring, maintaining and coordinating these multiple RFID readers and other sensors and actuator devices in RFID deployment, and for upgrading the firmware [6, 11].

Passive RFID systems have some common characteristics which are unique to the RFID domain [12]. These characteristics impose some significant challenges in the design of a device management part of RFID middleware as described below.

Limited communication bandwidth
RFID systems rely on the availability of unlicensed frequency bands with limited number of channels. For example, the 13.56MHz band has only one channel, and UHF frequency band in Europe allows for fifteen 200kHz-wide channels between 865.0MHz and 868.0MHz [13]. Readers need to listen for other transmitters using the channel before beginning to communicate with the tags [12]. Since large distribution centres might need to run as many as 100 readers, it is evident that readers need to co-ordinate their activities somehow to avoid reader interference and missing tags that pass by, while the reader is not operating. Another constrain is the limited bandwidth available per channel which limit the data transmission rate between readers and tags. Therefore, an efficient device management scheme is necessary to squeeze the maximum read rate out of larger RFID deployments [6].

Diversity of tag capabilities
Different RFID applications deploy different types of tags with different capabilities. For example, the memory on a microchip embedded in the tag usually contains a unique identifier but some microchips also feature small amounts of additional random access memory. Due to the increased power required to write to the EEPROM on the microchip, the maximum distance between reader and tag for a “write” operation is a fraction of that for a “read” operation [12]. Therefore, since RFID middleware is supposed to be application-agnostic, its design should take into consideration the diversity of tag capabilities.

Heterogeneous reader landscape

The diverse computational and networking capabilities of readers are also characteristic of RFID networks [12]. Low cost readers usually support only a single antenna and a serial RS232 interface. More sophisticated readers support several antennas, a TCP host interface, and ample computing resources for on-device data processing. For this reason, the device management scheme in use should be flexible and scalable to accommodate a diversity of reader capabilities and reader density.

Sensor and actuator support

In many applications it is not sufficient to only identify objects, but the current state of the objects in the physical world has also to be detected. For example, a perishable goods chain monitoring system should be able to monitor temperature data along the chain. The system has thus to provide the means to integrate sensors such as temperature, humidity or shock sensors and make their data accessible by the applications. In addition, in many applications it is not mandatory to operate RFID readers continuously due to the limited bandwidth available, and it is even undesirable to have readers transmit, while no tags are present. To initiate the tag inventory process at a reader when there are tagged objects arriving in the read range, external sensors, such as motion sensors, should thus be able to trigger the RFID readers [10]. In addition to sensors, applications often have to quickly interact with the physical world using different kinds of actuators such as locks or even simple traffic lights to signal an application state to an operator [19].

To address some of the challenges posed by the characteristics of RFID data and devices mentioned in this section, we intend to use intelligent software agents to develop an RFID middleware. In the next sections, we discuss our proposed RFID middleware using multi-agent systems.

3. MULTI-AGENT BASED RFID MIDDLEWARE SYSTEM

According to Jennings et al. [15], agents and multi-agent systems represent a new way of analysing, designing, and implementing complex software systems. The term of intelligent agent is used to denote a software-based computer system that enjoys the following properties: (a) autonomy (agents operate without the direct intervention of humans); (b) social ability (agents communicate with other agents); (c) reactivity (agents perceive their environment and respond in a timely fashion to changes that occur in it); (d) pro-activity (agents do not simply act in response to their environment, they are able to exhibit goal-directed behavior by taking the initiative) [16]. An agent has a set of goals, certain capabilities to perform actions, and some knowledge about its environment. To achieve its goals, an agent needs to use its knowledge to reason about its environment (as well as behaviors of other agents), to generate plans and to execute these plans.

A multi-agent system consists of a group of agents, interacting with one another to collectively achieve their goals. Among the benefits of applying agent technology to RFID middleware framework include intelligent decision making, easy interaction, efficient resource management, distributed system architecture, and interoperation among heterogeneous legacy systems.

3.1 System Architecture

Figure 1 shows the proposed RFID middleware architecture with four main layers. The first layer (lowest) deals with management of different types of sensor devices. The second layer is the data processing layer. Third layer constitutes the business process interpretation layer; it is responsible for associating the incoming observation messages from data management layer with existing business processes. The topmost layer of the architecture is enterprise application integration layer which integrates the RFID events with other enterprise applications.

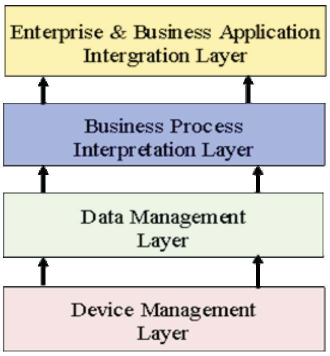

Figure 1: System Architecture

3.2 Functional Description

3.2.1 Device Management Layer (DeML)

This layer is responsible for coordinating multiple smart item devices. Beside RFID readers, these devices can include environmental sensors and PLC devices. In order to meet those requirements DeML consist of several sub-components, which includes Device interface, Device Monitor, Device configuration, Connection Manager, Device Profiler, Protocol Processor and PLC and Trigger handling. These sub-components are modeled and implemented as software agents.

Device Monitor agent is used to monitor the status of reader operations together with network connectivity. The Reader Monitor agent will display all pertinent health information of each reader connected on the system. Parameters to be monitored includes IP address, port number, connectivity status, operation status, antenna status, Reader time, tag read count etc.

The Device Configuration agent is responsible for network configuration together with persistent RFID-specific and device-specific configuration such as power level, active antennas, protocols, notification modes, trigger modes, frequency and administrative status.

The Connection Manager agent helps to ensure that the middleware has access to a particular RFID reader. RFID readers may be built in their own data exchange protocols and various communication interfaces, such as RS232, TCP/IP etc, therefore, the connection manager enables heterogeneous RFID readers from different vendors to interact uniformly with the middleware.

The Device Profile agent manages the outlined data about the devices which are deployed and registered in the middleware. For example, the profile data includes the device name, model number, Tag data protocols which the device support, frequency, network protocols, RF power, connectivity, and additional information for operating a particular device.

The PLC and Trigger handling agents are responsible for managing read triggering and other PLC signals, while the Protocol Processor agent is responsible for providing interoperability among multiple different types of readers and the middleware. It converts the commands issued by applications or users to reader-aware commands, and then passes them to the target reader.

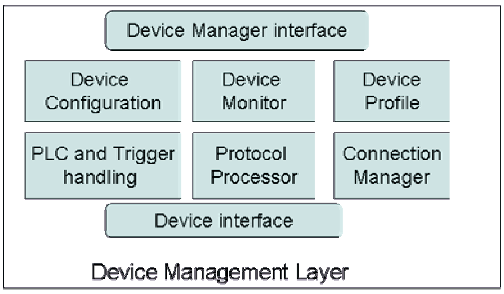

Figure 2: Device management layer with its functional components

3.2.2 Data Management Layer (DaML)

Problems such as multiple reads of the same tag, some tags not being read, erroneous reads etc; are addressed in this layer. This layer will contains several data processors such as filters, aggregators, loggers and buffers.

These data processors with appropriate processing rules and algorithms are responsible for verifying tag reads, filtering the incoming detected raw data, aggregating multiple incoming events into one higher–level event, temporary storage of observed events and translation of cleaned filtered and aggregated events into appropriate output format for RFID data exchange.

The DML has two main sub-components: the Processing Manager and Event Configuration. All the filters, buffers and loggers can be developed as units modeled as agents and they can be registered to the DML through the Processing Manager. This will allow users to register customised event filter units which satisfy application domain-specific requirements.

Event Configuration will display the registered processing units and allow user to change the filter, buffer and logger parameters.

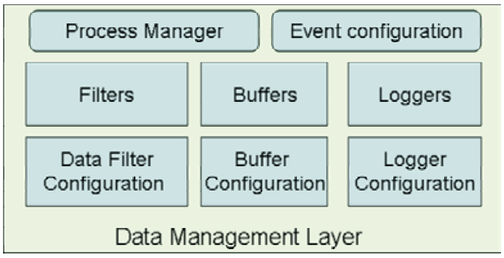

Figure 3: Data management layer with its functional components

3.2.3 Business Process Interpretation Layer (BPIL)

Issues related to extracting implicit information's i.e. meaningful business events from observed raw data and persistence data storage is addressed in this layer. The layer provides semantic data filtering, data aggregation, and support for decision-making. At this layer status and history information of tracked objects such as object location, aggregation information; inference and association rules together with information about the environment of a tagged object are maintained in a local repository. In this layer the incoming observed data from the lower layers are associated with existing business processes where by the data are aggregated, transformed and enriched with context information.

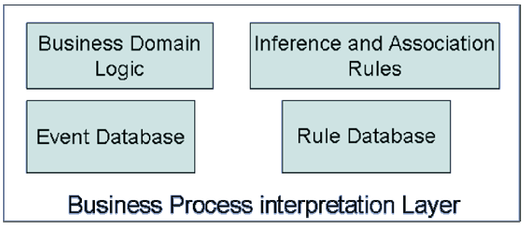

Figure 4: Business process interpretation layer with its functional components

3.2.4 Enterprise Business Application Integration Layer

This layer is responsible for integrating RFID data with other enterprise applications. It allows enterprise applications/users to subscribe to the information of interest.

The layer consists of service listener, message formatter, a common protocol for all connected devices, application configuration and interface configuration components.

The Service listener agent offer several communication protocols such as XML-RPC, SOAP-RPC, etc. It enables the middleware to receive request from different applications using different communication protocols. The Message formatter agent is used to transform the received request into the format understandable by the common protocol.

The common protocol agent enables an application or a user to communicate with a device without the need to know each device-specific protocol or API. The Protocol Processor agent in the DeML will then transform this common protocol into corresponding target device protocol.

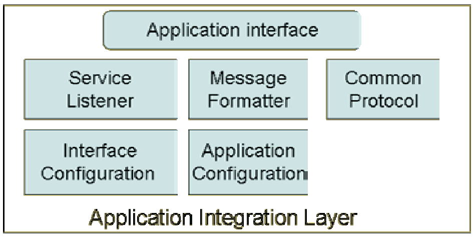

Figure 5: Business application integration layer with its functional components

4. RELATED WORK

The necessity of an RFID infrastructure and the application requirements towards such an RFID infrastructure has been discussed by several researchers [6, 11, 12].

Recently, major IT vendors are providing sophisticated RFID platforms, including the Sun EPC Network [18], SAP Auto-ID Infrastructure [11], Oracle Sensor Edge Server [19], IBM-WebSphere RFID Premises Server [20], Sybase RFID Solutions [22], and UCLA's WinRFID Middleware [21]. These platforms provide a general interface to collect RFID data from readers, and then forward the data to applications. These systems, however, mainly focus on identification and tracking of RFID tag information leaving the major task of detecting complex events to the user's application. Our system goes beyond this by including business domain knowledge within the middleware to achieve business process automation and semantic inference required to achieve operational effeciency and sustainable business processes.

5. SUMMARY

The widespread adoption of RFID with ultimate performance requires not only low cost tags and readers, but also appropriate software and architectural design. In addition, a justifiable return from RFID technology investments will only come from intelligent use of the data harvested from RFID systems. RFID middleware is the tool that helps to make sense of RFID tag reads, it translates the simple read data into useful information and propagate it to the appropriate enterprise information systems. In this paper we have discussed some common fundamental characteristics of RFID data and devices, which pose significant challenges in the design of RFID middleware system. We also presented a proposed multi-agent based RFID middleware architecture which addresses both data and device management challenges.

ACKNOWLEDGMENTS
The authors would like to thank DAAD and NRF for supporting this research work.

REFERENCES

[1] Smart Homes & Social Devices: RFID Takes off. The future of RFID: A memo series, Technology Horizon Program, Nov 2005, SR-926D. www.iftf.org

[2] Sharyn C. Leaver, Tamara Mendelsohn, Christine Spivey Overby and Esther H. Evaluating RFID Middleware: Picking The Right Solution For Integrating RFID Data into Business Applications. Yuen. August 13, 2004

[3] N. R. Jennings, K. Sycara, M. Wooldridge, “A Roadmap of Agent Research and Development”, Autonomous Agents and Multi-Agent Systems, 1, 275-306(1998)

[4] S. Chawathe, V. Krishnamurthy, S. Ramachandran, and S. Sarma, "Managing RFID Data," presented at 30th international conference on very large data bases (VLDB), Toronto, Canada, 2004.

[5] R. Derakhshan, M. E. Orlowska and Xue Li. RFID Data Management: Challenges and Opportunities.

[6] S. Sarma. Integrating RFID. ACM QUEUE October,2004.

[7] J. Brusey, C. Floerkemeier, M. Harrison and M. Fletcher. Reasoning about uncertainty in location Identification with RFID. *In Workshop on Reasoning with uncertainty in Robotics at IJCAI-2003, Acapulco, Mexico, 2003*.

[8] B. Carbunar, M. Ramanathan, M.Koyoturk, C. Hoffmann and A.Grama. Redundant reader elimination in RFID systems. Sensor and Ad Hoc Communications and Networks, pp 176 – 184, 2005.

[9] S. Jeffery, G.Alonso, M.Franklin, W.Hong, and J.Widom. A pipelined framework for online cleaning of sensor data streams. International Conference on Data Engineering (ICDE 06), page 140, April 2006.

[10] C. Floerkemeier, M. Lampe and C. Roduner. Facilitating RFID Development with Accada Prototyping Platform. 2006, www.accada.org

[11] C. Bornhoevd, T. Lin, S. Haller, and J. Schaper, "Integrating Automatic Data Acquisition with Business Processes - Experiences with SAP's Auto-ID Infrastructure," presented at 30th international conference on very large data bases (VLDB), Toronto, Canada, 2004.

[12] C. Floerkemeier and M. Lampe. RFID middleware design – addressing application requirements and RFID constraints. In Proceeding of Joint SOC-EUSAI conference, pp 219–224, Grenoble, Oct 2005.

[13] Radio Frequency identification equipment operating in the band 865 MHZ to 868 MHz. Technical Report EN 302 208-1, European Telecommunications Standard Institute (ETSI), 2004.

[14] M. Lampe and C. Floerkemeier. High-Level System Support for Automatic-Identification Applications.

[15] N. R. Jennings, K. Sycara, M. Wooldridge, "A Roadmap of Agent Research and Development", Autonomous Agents and Multi-Agent Systems, 1, 275-306(1998)

[16] M. Wooldridge, N. Jennings, Intelligent agents: theory and practice, The knowledge Engineering Review 10 (2) (1995) 115–152.

[17] A. Chella, M. Cossentino, and L. Sabatucci. Tools and patterns in designing multi-agent systems with PASSI. *WSEAS Transactions on Communications*, 3(1):352-358, Jan 2004. wake

[18] A. Gupta and M. Srivastava. Developing Auto-ID Solutions using Sun Java System RFID Software. Oct 2004. http://java.sun.com/developer/technicalArticles/Ecommerce/rfid/sjsrfid/RFID.html

[19] Oracle Sensor Edge Server. Oracle data sheet, 15/2/2006 http://www.oracle.com /technology/products/sensor_edge_server/collateral /Oracle_SES_Datasheet.pdf,

[20] IBM - WebSphere RFID Premises Server. http://www-306.ibm.com/software/pervasive/ws_rfid_premises_server

[21] UCLA WinRFID Middleware. www.wireless.ucla.edu/techreports2/WinRFID-Middleware.pdf

[22] Sybase RFID Solutions. http://www.sybase.com/rfid, 2005.