Agent-Based Control Framework for Mass Customization Manufacturing With UHF RFID Technology

Mengru (Arthur) Tu, Jia-Hong Lin, Ruey-Shun Chen, Kai-Ying Chen, and Jung-Sing Jwo

***Abstract*—Radio frequency identification (RFID) technology adoption in business environments has seen strong growth in recent years. Adopting an appropriate RFID-based information system has become increasingly important for enterprises making complex and highly customized products. However, most firms still use conventional barcode and run-card systems to manage their manufacturing processes. These systems often require human intervention during the production process. As a result, traditional systems are not able to fulfill the growing demand for managing dynamic process flows and are not able to obtain real-time work-in-process (WIP) views in mass customization manufacturing. This paper proposes an agent-based distributed production control framework with UHF RFID technology to help firms adapt to such a dynamic and agile manufacturing environment. This paper reports the design and development of the framework and the application of UHF RFID technology in manufacturing and logistic control applications. The framework's RFID event processing agent model is implemented in a smart end-point (SEP) device. A SEP can manage RFID readers, wirelessly communicate with shop-floor machines, make local decisions, and coordinate with other SEPs. A case study of a bicycle manufacturing company demonstrates how the proposed framework could improve a firm's mass customization operations. Results of experiments show the decentralized multiagent coordination scheme among SEPs outperformed the current practice of the firm in terms of reducing work-in-process and parts inventory.**

***Index Terms*—Mass customization manufacturing, multiagent system, radio frequency identification (RFID), RFID event processing agent (REA), smart end-point (SEP).**

I. INTRODUCTION

THE implementation of radio frequency identification (RFID) technology in industrial manufacturing and retail supply chain management has seen strong growth in recent years. This is partly due to Wal-Mart's RFID mandate to its suppliers [1]. As more companies along the global supply chain adopt RFID, RFID tags embedded can be expected to proliferate in virtually every industrial product, ranging from computers to automobiles, in the near future. Large retailers like Wal-Mart and government agencies such as the U.S. Department of Defense (DoD) have driven recent developments in RFID technology. This in turn has a diffusion effect on hundreds of suppliers and manufacturers as their products are required to be tagged before shipping to these giant customers. Individual consumers, on the other hand, are demanding more products and deliveries customized to their specific needs [2]. In light of changing customer preferences, firms must offer customized and personalized products for individual customers at mass production prices and delivery schedules. This phenomenon is called mass customization [3]. Mass customization changes the centuries-old tradeoff between tailoring a product to the needs of specific customers and the cost and time associated with delivering the desired product. Mass customization and customerization offer significant benefits to both customers and firms. The most significant benefits to the firm are a substantial reduction in inventory, the opportunity to enhance customer loyalty, and avoiding the pitfalls of commoditization [4]. To produce customized products efficiently, even at the level of item customization, products must be identified at the individual product level. RFID technology is currently the best solution to efficient customization as it enables firms to control individual items at all points of the supply chain, from manufacturing to distribution [2].

For firms making high-priced and highly customized products with large plant facilities, such as automobile or bicycle plants, combining good product identification technology with loosely coupled manufacturing application architecture enables flexible and agile manufacturing [5]. However, most manufacturing applications currently use a client-server computing architecture that uses static interfaces tightly coupled to the implementation of functions within these applications [6].

RFID technology provides a good alternative to automatically reading and writing product information. In addition to recording the identity of an object, RFID technology also documents its current status, recent past, and immediate future [7]. Using modern identification techniques, production systems can now produce variants of a product, or even different products, at a batch size of one [7]. A product with an RFID tag can be viewed as an intelligent product. Several studies in this emerging field indicate the necessity of adopting new manufacturing approaches for making intelligent products ([8], [9]).

Manuscript received November 25, 2008; revised March 30, 2009. First published September 01, 2009; current version published September 16, 2009.

M. Tu is with the Identification and Security Technology Center of the Industrial Technology Research Institute (ITRI), Chutung, Hsinchu 31040, Taiwan, and also with the Institute of Information Management, National Chiao Tung University, Hsinchu 300, Taiwan (e-mail: tum.iim95g@nctu.edu.tw).

J.-H. Lin is with the Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan (e-mail: d96921003@ntu.edu.tw).

R.-S. Chen is with the Department of Information Management, China University of Technology, Hukou Township, Hsinchu County 303, Taiwan (e-mail: rschen@iim.nctu.edu.tw).

K.-Y. Chen is with the Department of Industrial Engineering and Management, National Taipei University of Technology, Taipei 10608, Taiwan (e-mail: kychen@ntut.edu.tw).

J.-S. Jwo is with the Department of Computer Science, Tunghai University, Taichung 407, Taiwan (e-mail: jwo@thu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSYST.2009.2029663

Many manufacturing companies adapt new information systems to monitor manufacturing activities. These systems can take immediate action to resolve any emergent events that could disrupt production or cause customer dissatisfaction [10]. In other words, they change their business operations to provide product variety and customization through flexibility and responsiveness. These systems also remove data latency, analysis latency, and decision latency as much as possible [11]. Therefore, employing RFID technology and real-time business intelligence in mass customization, just-in-time production, and lean production enables a firm to achieve that goal and survive in today's hyper-competitive environment. To fully realize the potential benefits of RFID, firms must adopt new information system infrastructures that can better track and manage a large volume of distributed tagged objects within and between enterprises. On the other hand, this new RFID-based technology must also un-intrusively integrate with a firm's existing IT infrastructures to elevate its operational efficiency.

Previous research has addressed the integration of RFID technology with distributed intelligent control in mass customization manufacturing ([8], [9], [12], [13]). These studies clearly indicate that a distributed system is a better solution for RFID-based production control than a centralized one. However, a distributed architecture can introduce more shop floor dynamics than a centralized one. [14] shows that agent and multiagent based technologies have the power and flexibility to deal with such shop floor dynamics. Thus, agent-based solutions can play a key role in integrating RFID technology to better manage distributed manufacturing systems. Multiagent-based software platforms are usually endowed with distributed intelligent control functions, and are becoming a key control technology in new manufacturing control systems built in a distributed manner, such as intelligent manufacturing systems (IMSs) [15].

Researchers have conducted several studies on integrating RFID with agent technology in a manufacturing environment [16], [17]. These proposed RFID managing systems mostly use a centralized approach to process incoming RFID signals from RFID readers deployed around the plant. However, this architecture suffers from single point of failure and scalability issues. As more readers are added to this type of system, its performance may drop significantly. In addition, deploying a large number of readers on the shop floor is problematic since current industrial readers must be wired to a centralized computer. Therefore, this approach also suffers from a single point of failure and high deployment cost. The new agent-based framework is based on a distributed computing architecture. Ubiquitous computing technology can integrate physical objects and virtual information in an enterprise. A previous study shows that RFID technology can be used in ubiquitous computing applications [18]. An enterprise can naturally evolve into a ubiquitous organization if it can embed business logic in RFID readers and tags, making them smart devices that have the ability to make their own decisions and take appropriate action.

This paper helps develop the traceability, visibility, and control of mass customization manufacturing processes by integrating RFID and intelligent agent technologies. Instead of using only messaging and/or exchanging of software service to integrate enterprise systems, this study uses RFID to "hook" the physical objects in an enterprise to applications that are traditionally difficult to integrate. Specifically, this study uses UHF RFID technology to facilitate long range scans of multiple bicycle frames passing through large dock doors.

This paper proposes a distributed multiagent system framework for mass customization manufacturing called an RFID-based and agent-oriented distributed production (RFADP) system. This integration of RFID and agent technologies greatly reduces the complexities of managing dynamic production flows in mass customization, and improves the traceability and visibility of work-in-process (WIP) in manufacturing environments. This study proposes a multiagent system that makes it possible to respond to RFID events in real time, and studies a broad class of agent coordination strategies regarding just-in-time and just-in-sequence production. A case study of a bicycle manufacturing company demonstrates how the proposed framework can benefit a mass customization manufacturing plant. What distinguishes the proposed solution with other RFID-based and agent-based systems is that it includes a smart end-point (SEP). A SEP is an embedded device with built-in software agent components that allow the SEP to make its own decisions and coordinate with other SEPs. When connected to RFID readers and shop-floor machines, this embedded device becomes a smart reader capable of making local decisions in each processing station based on information carried by the tagged workpiece. This study presents a partial implementation of this framework, including a prototype of the smart end-point device, to illustrate the framework's decentralized intelligence architecture and show that the proposed approach can capitalize the potential of UHF RFID technologies. Finally, this study considers several criteria to evaluate the benefits of the proposed solution compared to the bicycle firm's current practice. These criteria focus on five aspects: tracking accuracy, due date compliance, labor cost savings, manufacturing throughput, and inventory control.

The remainder of this paper is organized as follows. Section II reviews the enabling technology. Section III analyzes a mass customization manufacturing that produces bicycles. Section IV details the system analysis and design of the RFADP system. Section V describes system implementation. Section VI presents case studies of applying RFADP to improve manufacturing processes. Section VII evaluates the experimental results of the RFADP system, and Section VIII provides final conclusions.

II. WIRELESS MANUFACTURING AND AGENT-BASED CONTROL TECHNOLOGY

A. *Background*

Recent advances in RFID technology have made many companies interested in applying RFID in their manufacturing plants. RFID accelerates the development of wireless manufacturing and several researchers have studied this emerging field ([19]–[21]). On the other hand, agent-oriented manufacturing and agent-based intelligent control for industrial manufacturing systems [10] have also started to reshape manufacturing paradigms. The following section reviews the literature on

RFID and agent technologies, which have a strong influence on the design of the proposed framework.

B. RFID Technology

Radio Frequency Identification (RFID) is a method of remotely storing and retrieving data using devices called RFID tags or transponders. An RFID system consists of hardware, such as RFID tags and readers, and software like RFID middleware. A RFID reader interrogates an RFID tag. The reader has an antenna that emits radio waves, and the tag responds by sending back its data. The middleware software usually runs on ordinary PCs or servers and provides an interface for many sensor technologies, including RFID, thereby achieving cross-platform hardware integration. Most RFID tags can be categorized as either active or passive tags. An active tag is powered by an internal battery and can typically function in read or write modes [22]. An active tag operates with up to 1 MB of memory, and has a greater reading range because of its internal power supply. A passive tag, on the other hand, does not rely on an internal power source. Instead, passive tags operate on power obtained from the transceiver device. However, passive tags have shorter reading ranges and require a higher-powered reader than active tags [23].

RFID supports three types of memory: read-only memory (ROM), read/write (R/W) memory, and write-once/read-many memory (WORM). A ROM tag, which is similar to a traditional bar code, comes equipped with a unique identifier after the purchase. R/W tags are more complicated than ROM tags and are more expensive because they can be written in increments, erased, and reused. Unlike R/W tags, WORM tags can be programmed only once. All three types of RFID tag are able to embed context-awareness (see [24], [25]).

Low frequency (LF) (125–134 kHz) and high-frequency (HF) (13.56 MHz) RFID systems are short range systems based on inductive coupling between the reader and the tag antennas through a magnetic field. Some manufacturing firms have already adopted LF or HF RFID technology in their production lines. However, LF or HF RFID technology cannot support plant-wide logistics and inventory control due to its limited reading range. Alternatively, ultra-HF (UHF, 860–960 MHz) and microwave (2.4 GHz and 5.8 GHz) RFID systems are long-range systems that use electromagnetic waves propagating between the reader and tag antennas. Though UHF has a considerably longer reading distance than HF and LF technologies, its performance in general is more susceptible to the presence of various dielectric and conducting objects, like metal or water, in the tag vicinity. The specialized design of a passive UHF tag antenna and its encapsulation material can greatly alleviate these problems. The longer read distances of UHF technology enable new use cases, such as scanning items as they pass through large portals such as dock doors [26].

Large organizations like Wal-Mart and the U.S. Department of Defense (DoD) have driven recent developments in RFID technology. However, they tend to adopt ultra-HF technology that ranges from 300 MHz to 2 GHz because this makes it possible to read passive RFID tags from up to twenty feet away. The primary RFID functional requirement of Wal-Mart and the DoD is the ability to read as many tagged objects as possible from a long distance to facilitate instant counting of products or real-time tracking of goods without line-of-sight. In such environment, UHF RFID technology seems to be a good choice for those purely logistic companies or organizations.

However, many large manufacturing firms have much more complex environments and harsh conditions. In some cases, HF RFID technology seems to perform better than UHF RFID due to its near field communication protocol. This characteristic makes HF technology less susceptible to interference from metal and water than UHF technology. As a result, many manufacturers have used HF technology to track workpiece status in the production line for some time. Nevertheless, a HF RFID reader can only read tags up to three feet away, and is limited to recognizing 100 tagged small objects at most (depending on the RFID reader capability). Besides, a HF RFID reader requires a large antenna to achieve its maximum reading range. These limitations make it impossible to detect larger workpieces like bicycles or automobiles from a distance, or to detect a stack of tagged products.

This study adopted UHF RFID technology instead of HF/LF alternatives for the following reasons. Firstly, UHF technology's longer reading range facilitates long range scans of multiple bicycles passing through large dock doors. Secondly, instead of using RFID tags for only production control and tracking purposes, our surveyed company, XMbike, plans to permanently implant a passive UHF RFID tag in its manufactured bicycles in the future. This tag will live through the bicycle's life cycle and the tag's content must be made accessible to supply chain partners around the globe such as retailers. The EPC global standard seems to be the most likely global supply chain standard in the foreseeable future, and this standard also adopts UHF technology. Finally, XMbike would like to adopt a single RFID frequency for both manufacturing and logistic purpose rather than having two different frequencies, HF and UHF, for their manufacturing and logistic operations, respectively. Using the same RFID frequency eliminates tag replacement and data conversion costs incurred by adopting two frequencies. To fulfill all three requirements mentioned above, XMbike opted for the UHF solution. Therefore, the prototype system in this study uses UHF technology, and takes the necessary measures to overcome its inherent constraints and weaknesses.

C. Agent Technology

The introduction of RFID technology to enterprise information systems is also causing an increased demand for a new kind of software system to process the continuing large influx of RFID data. In this context, agent technology is becoming a prime candidate to take on this new challenge. Since agent technology is an important area in artificial intelligence research, many studies on agent research offer various definitions of an agent. Russell and Norvig [27] defined an agent as anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators. Jennings and Wooldridge [28] defined an agent as a computer system situated in some environment that is capable of autonomous action in this

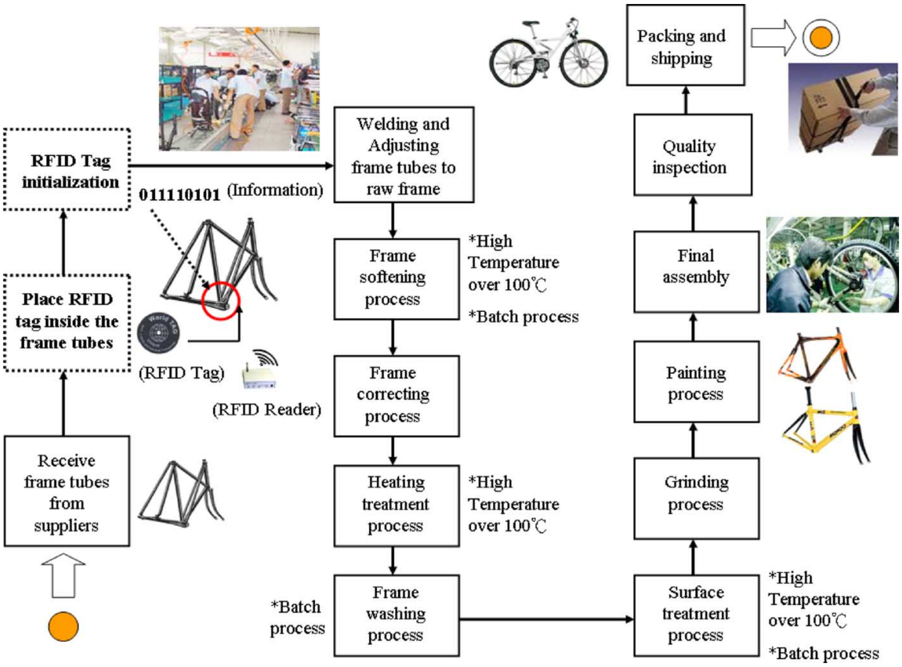


Fig. 1. XMbike manufacturing process.

environment to meet its design objectives. From the perspective of an agent's social ability, [29] believed that an agent is an active object that possesses certain capabilities to perform tasks. Further, an agent can communicate with other agents based on the organizational structure to cooperate the accomplishment of tasks. As for the application of agent technology, Jennings and Wooldridge [28] noted that agents are being used in an increasingly wide variety of applications, ranging from comparatively small systems such as email filters to large, open, complex, mission critical systems such as air traffic control. In real-time enterprise applications, agents facilitate real-time analytics to close the gap between business intelligence systems and business processes [30]. In manufacturing environments, multiagent systems are better suited than most current decision support systems to adapting to unanticipated disturbances [31].

III. SITUATION ANALYSIS OF MASS CUSTOMIZATION MANUFACTURING PRACTICE

This study examines a bicycle firm with mass customization manufacturing. This section analyzes its manufacturing processes and identifies key shop floor control system issues regarding the firm's current practice. Based on the operational scenarios of the company, this study develops and tests a solution called RFADP. The results provide the firm with a good reference on how to improve its current operations with new technologies.

A bicycle firm called XMbike (a fictitious name chosen to preserve the anonymity of the manufacturer) is a global leading bicycle manufacturing company. XMbike markets middle and high-end bicycles to Asia, Europe, and North America with manufacturing plants across Taiwan, China, and South East Asia. Fig. 1 gives a simplified sample model of the overall manufacturing process, from frame tube receipt to bicycle packing and shipping. XMbike has adopted the Japanese just-in-time (JIT) production system to manufacture their bicycle models. In addition, its shop floor control environment is loosely coupled and managed by field operators.

A. *Problem Statements*

Since XMbike use paper travelers with bar code labels to track and identify thousands of frames moving around the plant facility each day, current traceability is not accurate. They also use barcode labels to quickly and accurately guide pickers to the locations of the picked parts. However, keeping track of all parts in real time is still a daunting task that requires the company to employ many employees to perform frequent scanning of individual barcode labeled parts. Therefore, the current shop floor control and tracking processes are somewhat inefficient. Inaccurate and delayed WIP information may seriously impact the effectiveness of just-in-time and supply chain planning for XMbike, especially if it wants to adopt a mass customization production strategy. The following list identifies potential problems and inefficiencies in the current XMbike manufacturing process.

- The barcode system cannot be applied to processes like heat or surface treatment due to the harsh production environment in these processes. Therefore, the firm must employ manual tracking before the assembly stage.
- The Job Card system of data tracking contains only the product build information needed by the production operator at a particular station. The system does not have the capacity for real-time data tracking.
- Finding lost frames is time-consuming due to the large facility space and manual tracking process.
- In some production processes, such as heat treatment or painting, job cards must be detached from the bike frame during processing and then later reattached to the exact frame after processing is complete. All these actions must

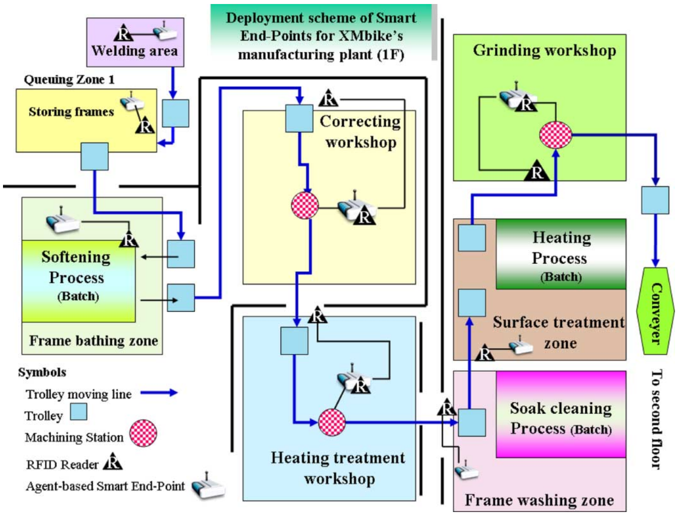


Fig. 2. Proposed RFADP system layout for XMbike -First Floor.

be performed by a field operator, and thus are prone to error.

- From time to time, rework processes take place on the first floor of the plant facility. Since frames in those areas are manually tracked, rework processes may lead to lost frames.
- Missing frames can delay and disrupt just-in-time and just-in-sequence assembly processes in the assembly stage. This is particularly true for a mass customization production plant.
- Real-time tracking of all components and parts is costly due to manual scanning of barcode labels for each part item.

B. Requirement Analysis

Though many of XMbike's plants are built for mass production of limited bicycle models, a number of plants are already manufacturing more high-end models in small lot sizes. These plants can make over 1000 models each year. Facing strong competition from other bike manufactures, XMbike plans to allow customers to select among a variety of model types, frame sizes, colors, and other features. The company estimates that customers can choose from about 6 million possible variations for their custom-made bicycles. The process employed to produce such custom-made bicycles requires not only highly trained and skilled workers, but also re-engineering of its information system to accommodate this change. Therefore, a mass customization manufacturing strategy naturally leads to the development of a system that rewards attention to detail and helps operators achieve "zero mistakes" in every step of the production process. In light of this strategic goal, XMbike plans to launch a mass customization production line within one of its most advanced plants. This plant currently manufactures a variety of high-end models with small lot sizes and special orders.

C. Feasibility Analysis – RFID Field Tests Analysis

Several field tests were conducted in one of the XMbike's high-end model manufacturing plants as well as in an Industrial Technology Research Institute (ITRI) laboratory to assess RFID capability and reliability in field manufacturing applications. Major testing criteria are: 1) RFID signal read ranges, which include fixed and moving tags; 2) environmental interference of radio signals, including heat, metal, and water; 3) the RFID tag position relative to the bikes (frames) and the readers; and 4) the amount of information that can be stored and read from the tags. These field tests use UHF RFID readers and C1G2 UHF RFID tags. Since some of the production processes have temperatures over 100°C (Fig. 1) and most of the bike frames are made of metal, a tag must survive in that environment for a period of time while remaining both heat resistant and free from metal interference. To resolve this issue, custom made heat-resistant and anti-metal passive read/write tags were purchased and placed inside frame tubes [see Fig. 1]. After the tags were placed in several testing frames, these frames proceeded through the entire manufacturing processes for about two days. Most of the tags survived the field tests, and testing statistics were collected in the experiment. XMbike was satisfied with the overall testing results of RFID tag performance in such a harsh manufacturing environment. The custom-made metal and heat resistant tags cost about $5 U.S. each, which is higher than many metal resistant tags on the market. However, XMbike considers it an acceptable price for its high-end customized bikes, which are priced anywhere from $3000 to over $10 000 U.S. each, especially if the tag is permanently implanted inside each bike.

IV. SYSTEM ANALYSIS AND DESIGN

Based on the nature of the previously mentioned bicycle manufacturing process, this study proposes an agent-based control framework utilizing UHF RFID technology and an agent-oriented approach for XMbike's mass customization manufacturing control system. This RFID-based and agent-oriented distributed production (RFADP) system employs RFID tags as physical connectors that integrate the nuts and bolts of physical objects and enterprise applications. The proposed framework also defines ontology for RFADP system to facilitate effective agent communication.

A. RFADP System Architecture

In light of XMbike's business model, this study presents an RFID-based and agent-oriented distributed production (RFADP) system. The RFID system infrastructure in the XMbike's plants should include RFID readers positioned at strategic points and RFID tags attached to the bicycle frames and boxes that contain key components. To manage RFID readers and shop floor operations, RFADP system deploys several smart end-point (SEP) devices to the production plant.

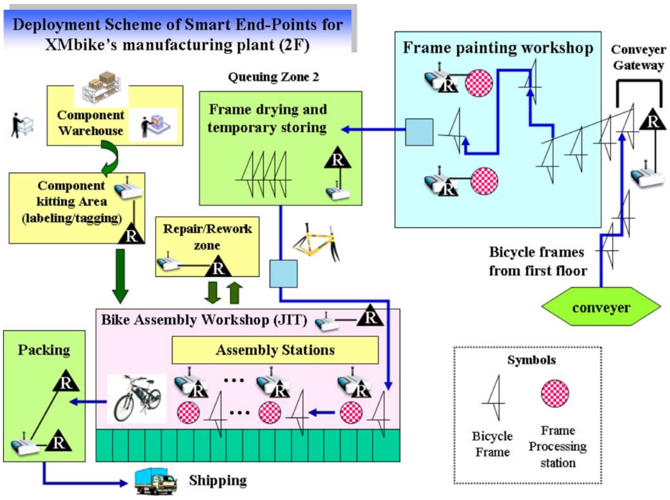


Fig. 3. The proposed RFADP system layout for XMbike -Second Floor.

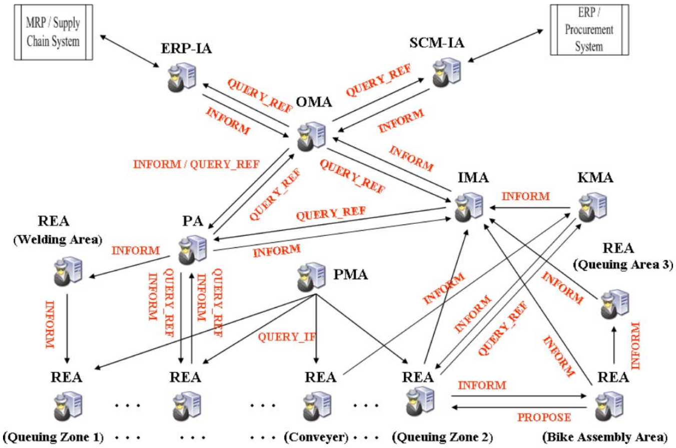


Fig. 4. Agent relationship and interaction diagram for the RFADP system.

Figs. 2 and 3 depict XMbike's factory layout and deployment scheme for RFID readers and SEPs within the plant.

B. Multiagent Organizational Structure and Agent Interaction Model for RFADP

The goal of this research is to develop a multiagent framework that is designed to offer intelligent collaborative support for just-in-time and just-in-sequence production strategies in a dynamic manufacturing environment. The following section defines eight types of agents and their models. Fig. 4 illustrates their interactions and functional behaviors. The communication protocol in Fig. 4 includes communicative acts defined by the Foundation for Intelligent Physical Agents (FIPA) [33]. The FIPA defined 22 communicative acts, some of which serve as a reference model for our agent communication protocol. Except for the RFID event processing agent (REA), where each instance of that agent resides in an embedded device called a SEP, the other types of agents are software programs with predefined goals deployed on a centralized server. The following list specifies the various agent types:

- *Order Management Agent (OMA)*: An order management agent prepares and generates work orders for workpieces. After work orders are generated, the order management agent relays order information to the product agent. Finally, the order management agent continues to track the order processing status of workpieces with the help of product agent.
- *Production Monitoring Agent (PMA)*: The production monitoring agent constantly monitors the production status based on the data polled from the REAs situated in each SEP device. This agent also issues alerts in case of machine breakdown or disruption of operation.
- *Inventory Management Agent (PMA)*: The inventory management agent collects all inventory information regarding components, WIP, and finished products in real time.
- *Kitting Management Agent (KMA)*: The kitting management agent informs operators when to prepare and label manufacturing parts with RFID tags for bicycle frames passing through the conveyer gateway on the second floor (see Fig. 3). This agent also constantly collects the component and part readiness information for frames staying in Queuing Zone 2 and provides real-time information on parts status to other agents upon request.
- *Product Agent (PA)*: The product agent can be further categorized into a product management agent and a workpiece agent. The product management agent is responsible for creating a virtual tag (an XML file containing detail build instructions for the workpiece) for each workpiece based on order information received from the order management agent. The product management agent then releases the virtual tag information to the REA residing in the first processing unit/workstation. The other task of the product management agent is to collect workpiece processing information from REAs and update the system database accordingly. On the other hand, the workpiece agent represents a corresponding workpiece in the factory. Usually, there are over a thousand workpieces (bike frames) in production in the XMbike plant on any given day. To simplify the implementation of SEP devices and fully utilize their capabilities, this study makes each SEP assume the role of a workpiece agent when it is processing a tagged workpiece instead of spawning a workpiece agent for each workpiece. In the proposed framework, a SEP can easily obtain information about a workpiece from the coding scheme of the RFID tag attached to the bicycle frame and the tag's corresponding virtual tag stored in the SEP. This approach eliminates the burden of creating and managing thousands of workpiece agents. A SEP can also represent a workpiece to negotiate with other SEPs regarding manufacturing service and scheduling.
- *RFID Event Processing Agent (REA)*: Depending on implementation requirements, each REA can represent a processing machine or a workstation (which includes more than one machine). This type of agent is deployed in a SEP device. Combining a RFID reader, middleware, and event processing logic in a single agent allows the REA to act like an intelligent reader that can make its own decisions even if the centralized server breaks down. Besides, it is easy to configure and deploy a REA in flexible manufacturing environment. With these characteristics, SEPs can form an intelligent reader network, as Figs. 2 and 3 illustrate.

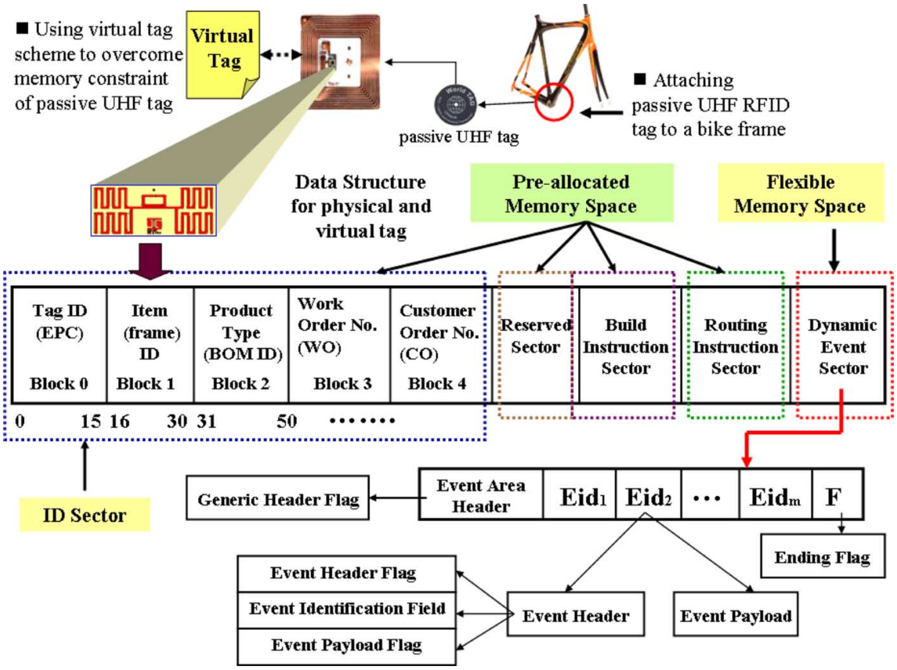


Fig. 5. RFID tag's data structure for the proposed RFADP system framework.

- ERP/SCM Interface Agents: These agents serve as interface programs for enterprise resource planning (ERP) or Supply Chain Management (SCM) systems, allowing the RFADP system to obtain necessary information from ERP or SCM systems.

C. Agent Ontology Model

Ontology, a concept borrowed from philosophy, is defined as the "science or study of being." In computer and information science, ontology refers to the specification of a conceptualized domain, and is often expressed in a computer interpretable format such as the XML. Ontology is used for agent's knowledge sharing and is becoming a crucial element in building a multiagent system. Only that which can be represented using ontology can be represented in an agent's knowledge base [34]. A good example of this trend is the recent work by [35] that describes a multiagent system that applies ontology and agent technology to construct a virtual observatory. According to the FIPA Ontology Service Specification [33], ontology is used for agents that want to converse and share a common knowledge for the domain of discourse. Java Agent Development Framework (JADE) [36] provides the content reference mode that defines all information elements in the discourse domain.

This study adopts an object-oriented design approach for ontology modeling and implementation. The following steps depict this ontology modeling and implementation process.

- This study begins the ontology modeling phase by conducting an extensive use case analysis of XMbike's manufacturing scenarios. Based on collected use cases and FIPA and JADE ontology specifications, we extract relevant domain concepts as objects.
- An ontology model is then constructed for the firm's manufacturing application based on these objects, their attributes, and the relationships between them.
- The ontology model is expressed in terms of a UML model and then transformed into a Java class diagram.
- This study then derives the E/R model from the Java class diagram.
- After confirming the Java class diagram and E/R model, this study starts the implementation phase.
- In the implementation phase, the object classes in the Java class diagram develop into Java objects (agent programs) which reside on a server or functional modules which dwell on SEPs. On the other hand, the E/R model is converted into relational tables. Part of the tag data structure in Fig. 5 contains data fields from a relational table featuring an intelligent product (a tagged product). Thus, designing the proper RFID tag data structure is a step in implementing the ontology model of XMbike's manufacturing process. The coding scheme for the tag data structure is based on XMbike's specific requirements. Fig. 5 illustrates the complete tag data structure scheme. This tag data structure can be divided into ID sector, reserved sector, build instruction sector, routing instruction sector, and dynamic event sector. Except for the dynamic event sector and routing sector, the rest of sectors are preallocated. The information stored in the build, routing, and event sectors can be interpreted by a tag ontology parser installed in each locally deployed SEP device.

D. Design and Implementation Issue of RFID Tag Data Structure

A set of standards or specifications, such as the ones developed by the EPC Global Organization, are applied to the design of the coding schemes for RFID tags. However, in addition to the 96-bit EPC-Code segment, many commercial UHF RFID

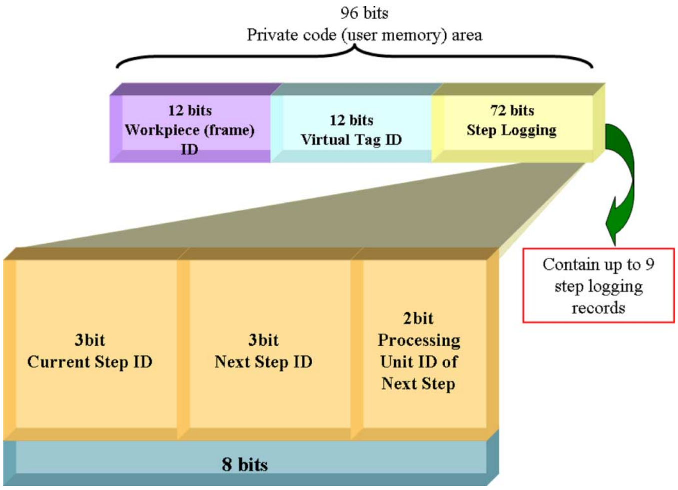


Fig. 6. Physical tag coding scheme.

tags currently available only contain a 128-bit private code segment (user memory) for user-specific data coding schemes. Although some have 512-bit private code segments, these may not be enough for the proposed tag data structure. Due to the memory capacity constraints of currently available UHF tags, we simply cannot store the entire data structure shown in Fig. 5 into a single tag memory at the moment. Therefore, this study proposes an alternative implementation scheme called a virtual tag, which borrows the virtual counterpart concept [37]. In this framework, the virtual counterpart is implemented as a virtual tag which can extend a physical tag's memory and accompany a physical tag as it travels along production lines. The virtual tag is a file that stores part of the proposed tag data structure to compensate for the limited memory capacity of the physical tag. The next section discusses the details of the design and implementation of both physical and virtual tags.

E. Encoding Scheme for Physical and Virtual Tags

To facilitate efficient read and write operations and preserve essential manufacturing process data on the physical tag, this study designs a coding scheme that contains two ID sectors. These sectors represent the virtual tag ID and the workpiece (bike frame) ID, respectively. A step logging sector represents a condensed version of the routing instructions and dynamic event sector. Fig. 6 shows this coding scheme, which encodes data in binary format. This tag structure is implemented on a private code area (96 bits) of a passive C1G2 UHF RFID tag in addition to a 96-bit EPC code segment. Since the private code area, or user memory, is mainly reserved for specific user needs, the coding scheme in this study is used for internal control only and not for inter-company supply chain transactions.

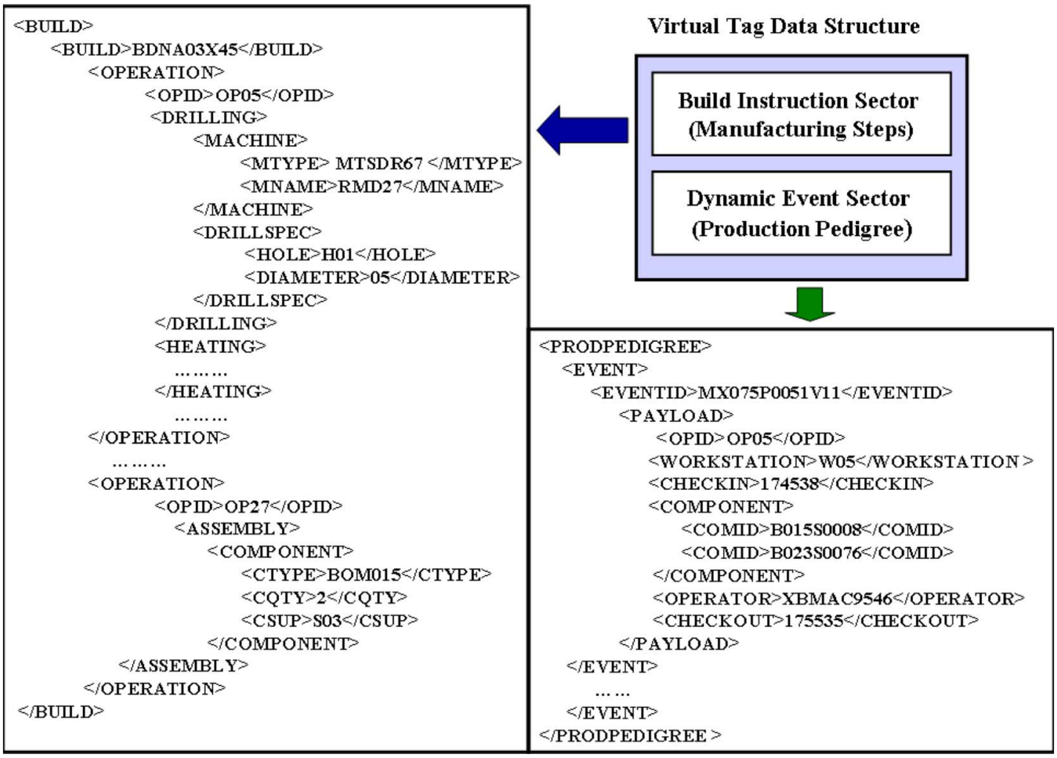
Due to the memory capacity constraint of most UHF tags, this study proposes a novel solution called "virtual tag" to compensate for the physical tag's memory constraints. Most of the tag data structure in Fig. 5 can be implemented on a virtual tag. A virtual tag is a file in XML format and can be exchanged among SEPs. Fig. 7 shows that the main construct of a virtual tag is divided into two parts. The Build Instruction Sector contains manufacturing steps that are interpreted by the SEP Process Engine. The Dynamic Event Sector can be viewed as a production pedigree. The proposed design of virtual tag and physical tag coding scheme not only overcomes the memory constraints of UHF tags, but also enhances system reliability. For example, when a virtual tag is accidentally lost during transmission, we can use physical tag information to recover virtual tag information from the hosting server up to the last server update, and then synchronize the information between the two. Likewise, when a physical tag is damaged, we can retrieve information from both the Build Instruction Sector and Dynamic Event Sector of the virtual tag to reconstruct a new physical tag to replace the damaged one. These mechanisms greatly reduce the risk of data transmission loss that could cause production line interruptions in a wireless and distributed production environment.

F. RFID Event Processing Agent Architecture

For a REA to efficiently process and manage RFID information, this study proposes a RFID event processing model and subsequently implements that model in an embedded device called a SEP as part of a RFADP prototype system. Some functions implemented in the model are similar to those in RFID middleware.

RFID middleware is an indispensable part of any RFID application because it can cope with the large influx of RFID signals. This middleware software must also handle physical reader management and raw RFID data filtering. RFID middleware decouples low level RFID data handling from application users and provides a logical interface for application users to manage physical RFID readers and [38]. Recent studies on RFID middleware architecture provide good insights and references for the REA design in this study. For example, [39] identifies which application requirements RFID middleware should meet, and addresses the constraints of passive RFID technologies regarding middleware design. An open source RFID middleware platform called Accda was developed to meet these application needs; [40] addresses issues facing an RFID network and develops a reader coordinator (RC) to deal with these issues. However, the goal of this study is to build a REA that not only acts as a micro version of typical RFID middleware, but can also be extended with functionalities that go beyond a middleware application. In this study, each REA is capable of processing RFID events, interpreting the proposed tag ontology model, taking proactive actions, and coordinating with its neighboring REAs. Fig. 8 shows the REA software modules in a SEP and RFADP applications in a hosting server. In actual SEP implementation, the Tag Handler, Session Controller, Service Dispatcher, and Event Manager are all encapsulated into a single module called the Process Engine. The following list describes in detail the role and responsibility of each module.

- **RFID Middleware Controller**: A middleware controller in REA is responsible for managing the physical RFID reader and preprocessing raw RFID data. This controller consists of filtering algorithms to handle the redundant or erroneous reads of RFID signals from the physical reader and mechanisms to facilitate efficient tag writing. The controller design and development is tightly coupled with the



```
<BUILD>
    <BUILD>BDNA03X45</BUILD>
        <OPERATION>
            <OPID>OP05</OPID>
            <DRILLING>
                <MACHINE>
                    <MTYPE> MTSDR67 </MTYPE>
                    <MNAME>RMD27</MNAME>
                </MACHINE>
                <DRILLSPEC>
                    <HOLE>H01</HOLE>
                    <DIAMETER>05</DIAMETER>
                </DRILLSPEC>
            </DRILLING>
            <HEATING>
              ………
            </HEATING>
              ………
        </OPERATION>
          ………
        <OPERATION>
            <OPID>OP27</OPID>
              <ASSEMBLY>
                <COMPONENT>
                    <CTYPE>BOM015</CTYPE>
                    <CQTY>2</CQTY>
                    <CSUP>S03</CSUP>
                </COMPONENT>
          </ASSEMBLY>
        </OPERATION>
</BUILD>
```

```
<PRODPEDIGREE>
  <EVENT>
    <EVENTID>MX075P0051V11</EVENTID>
      <PAYLOAD>
        <OPID>OP05</OPID>
        <WORKSTATION>W05</WORKSTATION >
        <CHECKIN>174538</CHECKIN>
        <COMPONENT>
          <COMID>B015S0008</COMID>
          <COMID>B023S0076</COMID>
        </COMPONENT>
        <OPERATOR>XBMAC9546</OPERATOR>
        <CHECKOUT>175535</CHECKOUT>
      </PAYLOAD>
  </EVENT>
    ……
  </EVENT>
</PRODPEDIGREE >
```

Fig. 7. Virtual tag data structure.

brand of RFID reader in question, since the implementation of tag read/write functions must be based on the application programming interface (API) provided by the reader's vendor. After performing RFID data preprocessing, the RFID Middleware Controller relays a single set of tag data string to the Tag Handler.

- **Tag Handler (Ontology Parser)**: The Tag Handler is responsible for processing incoming tag strings from the RFID Middleware Controller and issuing tag read/write commands to the controller. It is also responsible for receiving virtual tags sent from other agents. A built-in tag ontology parser interprets the coding scheme of the physical tag and generates RFID events that will trigger the REA to take further actions. The Service Dispatcher then handles the virtual tag. In a manufacturing environment, RFID events can be categorized into two categories. The first one deals with complex event processing, and usually involves machining processes or assembly operations. The second one handles much simpler event processes like entering (exiting) a workstation, or passing a RFID-based dock door. In this study, the SEP that deals with the first type of RFID events is considered a "Stateful Reader" while the SEP that deals with the second type of RFID events is considered a "Stateless Reader." Most modern RFID readers are deployed in a logistics environment, and are therefore stateless readers. A complex manufacturing environment, however, requires both types of reader. Considering that there are two major types of RFID events in the XMbike case, this study includes a sub-module in the Tag Handler called "Pure-Tag-Logging" to deal with the second type of events, which do not require further processing by the Process Engine. Pure-Tag-Logging reads only the tag ID of multiple RFID tags and logs the ID information along with other parameters in a predefined storage area. For example, some workshops do not deal with machining and assembly processes, but with batch chemical processing jobs such as XMbike's soak cleaning processes. These operations can be categorized as simple events, and the SEPs deployed in those areas should employ Pure Tag Logging accordingly. SEPs are typically preconfigured to their appropriate types before deployed to the shop floor.
- **Session Controller**: The session controller is responsible for event session control and temporary data caching services. It spawns a new event handling session that enables the Process Engine to handle a workpiece processing event passed on by the Tag Handler. It then caches session data during processing and closes the session when the processing cycle for a workpiece is complete. The session controller helps a REA make sure that it is working on the same workpiece in an event handling session. The session controller also helps improve the processing speed and data integrity of each workpiece processing event. The session control mechanism of the session controller also allows a REA to process multiple tagged workpieces at the same time. For example, in a batch assembly task, the session controller can spawn a unique control session for a group of detected workpieces and keep the tag ID of each workpiece in that session. After assembling each workpiece, the REA can read each tagged workpiece in the output buffer and match their tag ID numbers with those cached in the session. The REA can only

update the state information on matching workpieces. This further reduces processing errors in applying RFID to batch assembly operations.

- **Service Dispatcher**: The major tasks performed by the Service Dispatcher include locating virtual tags for the corresponding RFID event, parsing virtual tag content, selecting recipes and actions from the recipe and action library, and invoking recipe and action services. Service Dispatcher also collects feedback information from the results of these executed services.
- **Event Manager**: The Event Manager is responsible for preparing new events and producing a write or clear instruction to the Tag Handler. The Event Manager determines the next operation procedure and processing unit based on virtual tag information. It then generates a new event code (the step logging section of Fig. 6), and tells the Tag Handler to write the event code to a tagged workpiece. Before instructing the Tag Handler to write a tag, the Event Manager may consult with the session controller and Tag Handler to determine whether or not the REA is still working on the same tagged workpiece.
- **Recipe and Action Library**: Recipes and actions are executable business logic components encapsulated in dynamic-link libraries (DLLs). These components are first developed and tested by engineers, and then deployed to SEPs from the hosting server. Recipes are specific function calls that can invoke corresponding programmable logic controller (PLC) programs, computer numerical control (CNC) programs, or other operational procedures. Actions, on the other hand, are business logics that model the agent behavior or internal control logic of a SEP
- **Agent Coordinator**: The Agent Coordinator facilitates interactions among REAs as well as interactions between REAs and other server-side agents. Action components form the primary construct of the Agent Coordinator. Later sections in this study will illustrate agent coordination scenarios.
- **Shop Floor Interface Module**: This module serves as a communication channel that relays information to, and accepts feedback from, the external environment, which may include machines or user interface consoles.
- **Client Side shop floor control console**: This is a user interface console that facilitates shop floor operations and monitoring.
- **Local Storage**: This persistent repository resides in a SEP that connects to a machine or workstation on the production line. The following types of information are kept in the local storage as files.
  - Master data for each specific workstation on the production line.
  - System parameters and information specific to the local workstation.
  - Virtual Tag and its processing information; they are temporarily stored here.
  - Tag transaction logs; these logs are temporarily stored here and periodically relayed to the database in hosting server.

G. *RFID Event Processing Model*

The key components involved in RFID event processing and ontology interpretation are embedded in the RFID Middleware Controller, Tag Handler, Session Controller, Service Dispatcher, and Event Manager. Fig. 9 depicts their processing steps and information flows as a block-like representation of interactive components (BRIC) model.

This study uses BRIC formalism ([31], [32], [41]) to model the REA's internal state and its interaction with other agents within the agent environment. We map RFID event processing interaction steps to "place" symbols of BRIC model illustrated in Fig. 9. Place symbols are categorized in two types, denoted as either a conventional Petri Net place or an input/output communication place; the former represents an agent's internal methods and actions while the latter represents the communication among agents and communication between agents and their environments. The following section describes the places (P01 to P21) for the proposed RFID Event processing model.

- P01: The SEP Tag Handler receives a virtual tag released from another SEP or server and saves it in local storage. This virtual tag may come from an upstream or downstream SEP, depending on whether or not that the SEP adopts a push or pull production mode.
- P02: The Tag Handler receives a RFID tag string from the RFID Middleware Controller and immediately parses that tag string to obtain tag ID and other tag parameters (ex. virtual tag ID). The SEP can start to process a tagged workpiece only when the physical tag has a matching virtual tag.
- P03: The Tag Handler generates a new RFID event for the tagged workpiece.
- P04: The Tag Handler invokes the Session Controller to spawn a new event handling session for the RFID event of the tagged workpiece.
- P05: The Tag Handler invokes the Service Dispatcher to locate the virtual tag for the corresponding RFID event for this new session. If no virtual tag is found, the Tag Handler issues an alert to the Shop Floor Control Console.
- P06: Once a virtual tag is found, the Service Dispatcher parses the virtual tag to get all instructions and specifications regarding a workpiece.
- P07: Based on the obtained instruction and specification codes, the Service Dispatcher selects appropriate recipes and actions from the recipe and action code-book library.
- P08: The Service Dispatcher prepares service requests for any external servicing program specified in the recipe as well as any actions required by this event handling session.
- P09: The Service Dispatcher forwards service requests and parameters to external servicing programs (residing in the processing unit/machine or operation console) and hands control over to those programs.
- P10: The Service Dispatcher collects feedback (execution results) from both external servicing programs and internal actions.
- P11: The Service Dispatcher informs the Event Manager about the completion of tasks and also relays feedback information to the Event Manager.

- P12: The Event Manager determines the next operation procedure and processing unit for a workpiece. Determining the next processing step for a workpiece is quite straightforward, as the information can be obtained from the virtual tag. However, if there is more than one processing unit of the same type (ex. a tool/machine group), the Event Manager may need to invoke the Agent Coordinator to negotiate with the SEPs of the available processing unit to secure the best deal. The negotiation mechanism may involve multiagent auction or bargaining strategies. However, the negotiation mechanism is currently beyond the scope of this paper and thus we will not elaborate further.
- P13: The Event Manager initiates event updating processes.
- P14: The Event Manager prepares new step logging data. Fig. 6 illustrates its data format.
- P15: The Event Manager sends new step logging data to the Tag Handler and instructs the Tag Handler to write this information on a physical tag. Additionally, the Event Manager may request the Tag Handler to obtain the tag ID of a workpiece from the middleware controller and compare that ID with the tag ID already cached in the current session to see whether they are the same. This extra procedure helps the REA make sure it is working on the same workpiece, and thus safely updates information on the tagged workpiece accordingly.
- P16: The Event Manager confirms the RFID reader's writing operation has successful completed.
- P17: The Event Manager initiates a virtual tag updating process and informs the Session Controller to perform that operation.
- P18: The Session Controller stores workpiece processing information on the virtual tag and closese the session, thus releasing the cached event data of the workpiece and other computing resources from the SEP. Alternatively, depending on the company's policy, the Session Controller may need to synchronize workpiece processing information with the system database on server.
- P19: The Agent Coordinator receives a processing request from another SEP.
- P20: The Agent Coordinator selects an appropriate virtual tag to forward (in case there are multiple virtual tags)
- P21: The Agent Coordinator forwards a virtual tag to the SEP initiating the request.

V. SYSTEM IMPLEMENTATION AND DEMONSTRATON

To validate and prove the applicability and usefulness of the proposed framework for XMbike's operations, this study implements a prototype system, including software components and hardware devices, to exhibit the proposed design.

A. Development Environment

The development environment for the prototype system is described below:

- Operating Systems: hosting server: window XP; client SEP: window CE for the embedded system.
- Database: SQL Server (Server).
- Programming language: Java technology stack (JSP, JDBC, Java beans…etc.) for server side agents and applications; Microsoft C# for the REA.
- SEP Device: embedded system with ARM CPU architecture.
- Wireless Communication Protocol: TCP/IP and Bluetooth.
- RFID Tag: UHF RFID tag (EPCglobal Class 1 Gen 2 tag).
- RFID Reader: UHF RFID Reader (vendor: Tagsense).
- Factory Emulation Tool: Lego controllers and components.

B. Test-Bed Environment

Based on the XMbike plant environment and the proposed system architecture, this study integrated several SEP devices with Lego components and controllers to build an automated production line as the test-bed. This platform can simulate the fabrication and production of bicycles, motorcycles, or cars in real world scenarios. This test-bed presents an integrated information system used to control the shop floor environment, provide real-time tracking and tracing of WIP, and facilitate the adaptive control of dynamic mass-customization manufacturing processes. Fig. 11 shows a bird's-eye view of the test-bed layout, which is used for concept proving of the proposed framework.

C. Implementations

This section first discusses the implementation of a SEP. Fig. 10 shows the SEP system architecture. A SEP must be equipped with a RFID Reader to read/write information from/to tagged workpieces and products. The SEP interacts with external machine or user interface through a hardware I/O interface. A network device with a LAN or WLAN module helps the SEP to communicate with other SEPs or devices. Finally, an embedded operating system and software components of REA enables a SEP to perform complex tasks.

A SEP basically implements most of the components mentioned in the REA architecture in the previous section. It combines the Tag Handler, Session Controller, Event Manager, and Service Dispatcher into a single process engine. The process engine then uses recipes and actions as executable business logic components to perform certain tasks for the SEP. The Message Queue Handler (MQH) module built on the Microsoft Message Queuing (MSMQ) technology enables the SEP to wirelessly communicate with other SEPs and hosting servers. Within MQH, a sub-module that implements Bluetooth technology facilitates wireless communication between the SEPs and Lego controllers. A Lego controller with its controlling device emulates a CNC machine in real manufacturing environment. Persistent data storage of SEP is implemented as a file system. The programming language used in SEP development is C#.

The backend part of the system consists of server side agents and two subsystems to take care of specific system operations (see Fig. 8). The roles and responsibilities of server side agents are described in Section IV. Each server side agent is endowed with business logics to perform certain tasks. These agents are not developed with a JADE software platform in this research; they are currently implemented as JAVA objects.

Two subsystems are also implemented in JAVA. The SEP management subsystem provides system configuration and SEP

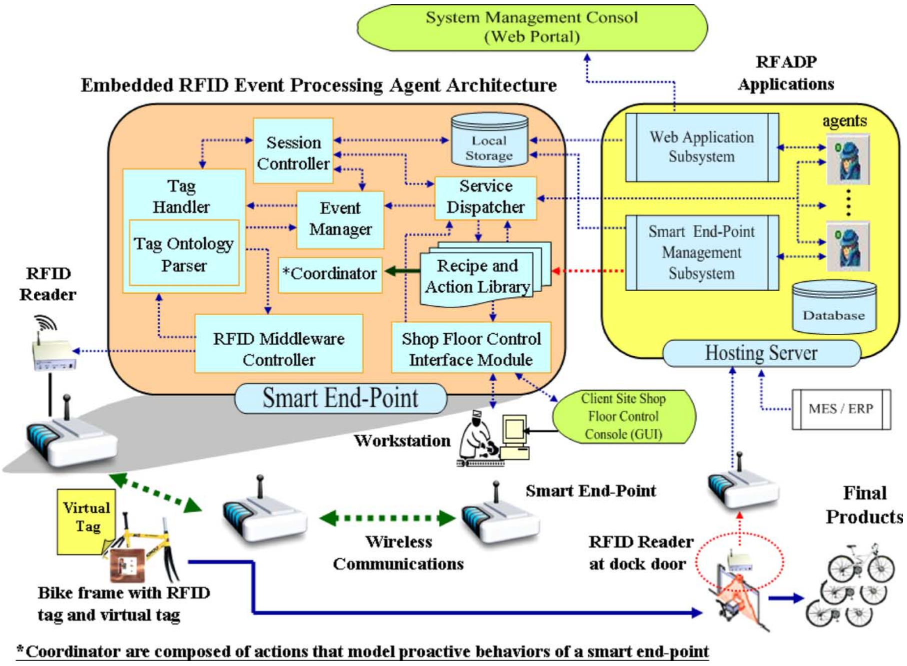


Fig. 8. REA and RFADP system architecture.

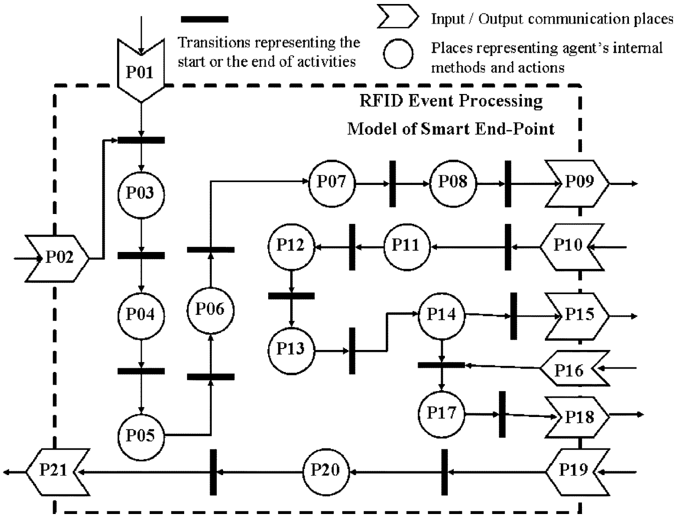


Fig. 9. BRIC model for the RFID event processing of REA.

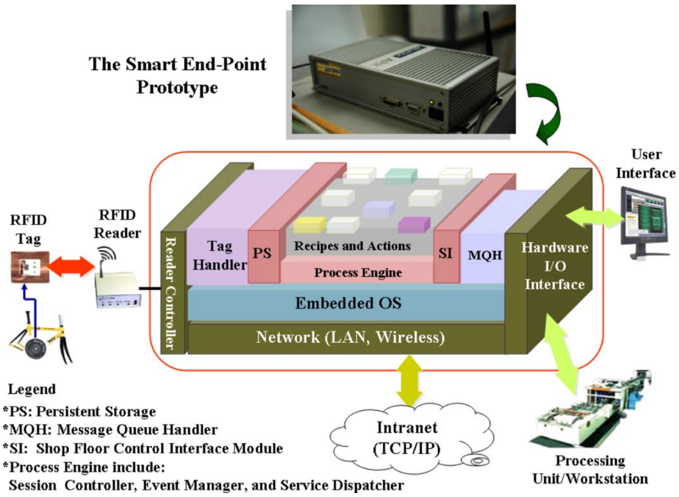


Fig. 10. SEP architecture.

health monitoring services; this subsystem also helps deliver actions and recipes (DLLs components) to the SEPs. The web application subsystem supports the operation of the web portal. Finally, to enable the server applications to communicate with the front-end SEPs, the hosting server contains a customized MSMQ component to facilitate two-way communications between server applications and SEPs.

VI. CASE STUDY

A. Scenario I – Workshop Operations

The test-bed simulates the shop floor operations of the XMbike manufacturing plant. The facility layout consists of three areas: one queuing zone and two production areas, as shown in Fig. 11. A production supervisor releases the WO in the form of a virtual tag from the SEP management subsystem to the SEP of the painting workstation. A workpiece is then fitted with a passive UHF RFID tag preloaded with the aforementioned tag data structure. Once the tagged workpiece has been placed on a workstation, a series of processing events are triggered and executed as described in Section IV. Operators can easily operate the shop floor control console, and senior managers can access the real-time information regarding the whereabouts of all the workpieces and the current status of each workpiece through a real-time web portal. This demonstration shows that the proposed framework virtually eliminates the manual barcode scanning and run card (paper travellers) tracking currently used by XMbike. The proposed approach will help the company achieve near 100% visibility of its manufacturing process and

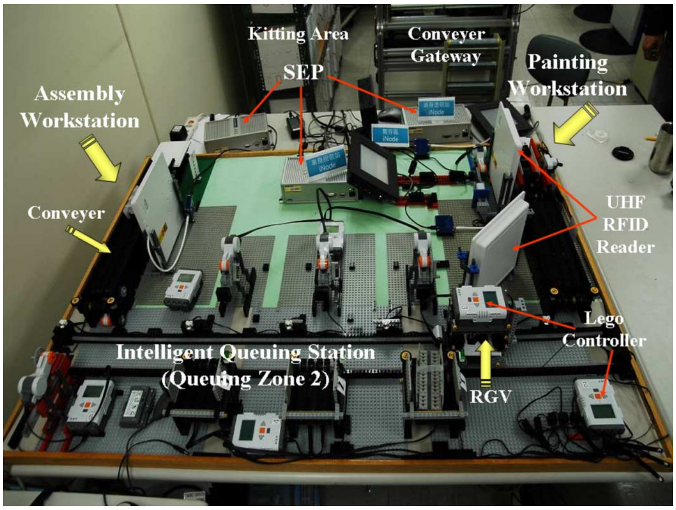


Fig. 11. Bird's-eye view of the test-bed layout.

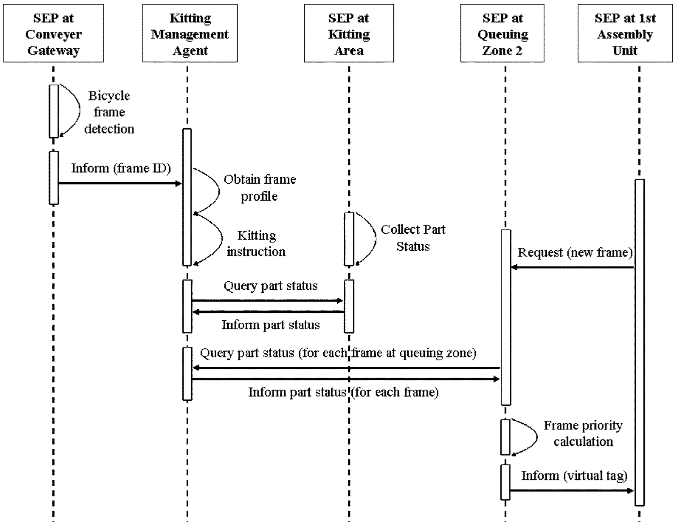


Fig. 12. Multiagent coordination processes.

WIP information, which is not possible under current control and tracking practices.

B. Scenario II – MultiAgent Coordination

As mentioned in Section III, XMbike has adopted the just-in-time (JIT) and just-in-sequence (JIS) production mechanisms to keep its inventory at a minimum and reduce the manufacturing lead time. To achieve these goals, parts must be prepared in advance and arrive at the assembly line just when a specific model of bike frame needs them. If a bike frame scheduled for the assembly line is somehow lost in the plant, those parts could become excess inventory. The problem of missing frames occurs from time to time, and the company must employ extra effort to tackle this problem. To help XMbike overcome this issue, this study offers solutions that utilize multiagent coordination capabilities to improve tracking accuracy of bike frames and facilitate adaptive assembly scheduling. Fig. 11 shows the layout of the test-bed. The SEP located at the conveyer gateway of the second floor constantly detects incoming bike frames from the first floor (Figs. 3 and 11), Kitting Management Agent (KMA) is being informed of each incoming frame in real time. This action is preloaded in the SEP of conveyer gateway. The KMA can then obtain frame profiles and parts information for each frame from the system database. Finally, the KMA instructs operators to prepare the appropriate components and parts. When the operators finish the parts kitting, they place the parts in a box and label the box with a passive RFID tag. These boxes are then stocked in the component kitting area (Figs. 3 and 11). Once the SEP deployed in the component kitting area detects new information about the status of the latest part, it immediately relays the information to the KMA, as illustrated in Fig. 12.

The bike assembly workshop (Figs. 3 and 11) uses the Kanban mechanism to achieve the pull strategy required for JIT production. The most important feature of the Kanban mechanism is that using Kanban cards as pull signal the downstream workstation order the amount of workpieces from its upstream workstation only what has been consumed by the downstream workstation. This eliminates waste and increases the ability of the production floor to respond to changing customer demands. The proposed framework portrays the multiagent decision model as a cooperation mechanism among SEPs and server side agents. This cooperation mechanism simulates the Kanban mechanism mentioned above. In the XMbike plant, the assembly workshop pulls bike frames from Queuing Zone 2. As a result, Queuing Zone 2 usually holds a predefined optimal quantity of painted fames to serve any request from the bike assembly workshop. Once the SEP of the first workshop assembly unit assembles a bike frame, it initiates a request to the SEP in queuing zone 2 for a new frame. The SEP in Queuing Zone 2 then checks the latest part status for each bike frame ready for assembling with the help of the KMA. It then calculates the frame priority based on its due day and its part readiness status. Finally, the SEP in queuing zone 2 selects the bike frame of highest priority and delivers the virtual tag of that frame to the SEP in the first assembly unit. The Lego conveyer then pulls that frame to the assembly workshop. Fig. 12 illustrates the aforementioned multiagent coordination processes.

VII. EVALUATION

A. Performance Evaluation of the Prototype System

To evaluate the prototype system, this study conducted experimental tests on the performance of a single SEP device and the effectiveness of the multiagent coordination scheme among multiple SEPs. Table I and Fig. 13 show these results.

Table I divides the field tests of SEP performance into four parts. The average response time for event preprocessing represents the time it takes a SEP to complete processing tasks from P01 to P09, just before handing control over to a Lego workstation. On the other hand, the average response time for event post-processing measures the time it takes a SEP to complete tasks P10 to P18. This study uses the following basic notions to measure SEP system reliability [42]: mean time to failure (MTTF) defines the average time to the next failure; mean time to repair (MTTR) is the average time it takes to diagnose and correct a fault, including any restart times. For example, in a 12 h test run, a SEP could jam four or five times, and thus its MTTF would be about 166 min. For a SEP, its MTTR is generally the

TABLE I
SEP FIELD TEST PERFORMANCE RESULTS

Average response time for event pre-processing	**Average response time for event post-processing**	**Average crash frequency**	**Average startup time**
0.8 (second)	**3 (second)**	**9 (day)**	**6 (second)**

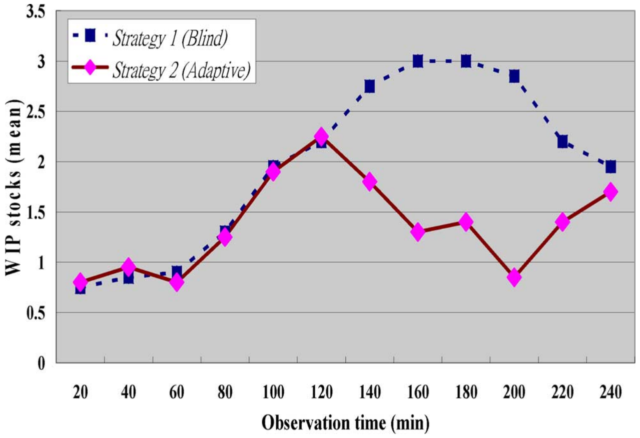


Fig. 13. Comparison of two WIP managing strategies adopted by intelligent queuing station at test-bed.

time it would take to restart. It takes about 15 s, on average, to restart a SEP once it is jammed. In light of these performance results, the next planned phase of SEP development will devote more resources to improving the system's reliability.

To test the performance of multiagent coordination, we set up the experiment on a test-bed consisting of two Lego workstations, each equipped with a SEP device, and one Lego-based intelligent queuing zone (queuing station), equipped with a SEP device as well. Fig. 11 shows the test-bed layout. The intelligent queuing station emulates the queuing zone 2 of XMbike's facility, while the two test-bed workstations emulate the Frame Painting Workshop and Bike Assembly Workshop, respectively. Since the queuing station only has three slots to hold workpieces, the maximum capacity of the queuing station is three workpieces. The Lego painting workshop is configured to process workpieces at a constant rate while the Lego assembly workshop is configured to pull a workpiece from queuing station and process it at different speeds at different time intervals in the experiment. The test-bed also performs two WIP managing strategies separately under the same conditions. We only need to change the business logic components of SEP in the queuing station to fulfill the simulation needs of the two WIP control strategies. The performance indicator is the mean value of WIP stocks during each 20-min time interval. The queuing station SEP conducts WIP counting every minute. Strategy 1 simulates XMbike's current practice, which we call a blind pull strategy. Whenever the queuing station is full ($\text{available slot} = 0$), the queuing station notifies the Lego painting workshop to stop working. The Lego assembly workshop decides to pull a workpiece from the queuing station at its own pace regardless of Strategy 1 or 2. On the other hand, strategy 2 is an adaptive strategy that simulates multiagent coordination based on the following scheme.

- The initial threshold keeps two WIP stocks in the queuing station. When the queuing station has two slots available, the SEP will send a pull signal to the Lego painting workshop (upstream workstation).
- Whenever the buffer stocks are down to zero, the SEP must immediately send a pull signal to the upstream workstation.
- The REA calculates the average queuing stock each time the queuing station receives three consecutive pull signals from the Lego assembly workshop (downstream workstation). If the average queuing stock is 50% greater than the previous threshold, the REA reduces the threshold by one unit. Conversely, if the average queuing stock is less than the previous threshold by 50%, then the REA increases the threshold by one unit
- The triggering condition for the queuing station to stop the upstream workstation (painting) from working is $[\text{queuing station available slot} < (\text{buffer capacity} - \text{threshold})]$.
- The triggering condition for the queuing station to notify the upstream workstation (painting) to start working is $[\text{queuing station available slot} > (\text{buffer capacity} - \text{threshold})]$.

Fig. 13 shows the results of one experimental run. Initially, the pulling speed of the Lego assembly workshop was kept at a constant rate and then gradually slowed down between time intervals 80 and 180. During the last phase, it picked up speed between the 180 and 240 time intervals. These statistics reveal that the adaptive strategy clearly outperformed the blind strategy in terms of mean WIP stocks. This experiment also illustrates that multiagent RFADP coordination helps reduce the inventory of work-in-process, which in turn leads to a reduction in parts inventory.

B. *Evaluation of Benefits*

As mentioned above, XMbike currently uses paper travelers (run cards) with bar code labels to track and identify thousands of workpieces (bicycle frames) and parts moving through its production facility each day. Tracking and tracing its manufacturing process information involves human intervention that often causes errors and delays in obtaining information. These problems also affect process quality control. Thus, the firm would like to evaluate the effects of the proposed solution in the following five areas: tracking accuracy, due date compliance, labor cost savings, manufacturing throughput, and inventory control. Table II lists these evaluations details, and the following section explains these evaluation metrics.

Workpiece (frame) tracking accuracy calculates the percentage of correct information in the firm's manufacturing execution system (MES) regarding a bicycle frame's where-abouts in the plant. Currently, three out of ten records concerning the frames' whereabouts are incorrect. The workpiece (frame) loss rate

TABLE II
ESTIMATED IMPROVING EFFECT IF THE PROPOSED RFID-BASED MANAGEMENT CONTROL SYSTEM IS ADOPTED

Evaluation of benefits (performance metrics)	**Status Quo**	**Estimated improving effect after 2 years**
Workpiece(frame) tracking accuracy	< 70%	>95%
Workpiece(frame) loss rate (misplacement and transaction errors)	6%-7%	<1%
Due date missing rate for bikes sold (yearly)	5%	<1.5%
Labor cost savings for component assembly (in terms of matching right part to right frame)	---	10%-30%
Labor cost savings for tracking workpieces (frames)	---	20%-40%
Labor cost saving for inventory audits (for parts, frames in production, and deliverable bikes)	---	25%-50%
Reduction of throughput time for making a customized bike	---	40%-50%
Reduced Inventory (includes frames and components/parts)	---	10%-15%

tracks the percentage of bike frames in each working day that are lost due to misplacement and transaction errors. The rate of bikes that miss their due dates is calculated as the number of bikes shipped to the customer after the due date over all bikes sold that year. The throughput time for making a bike measures the total manufacturing and processing time for a customized bike: it includes the time to receive an order, prepare parts, manufacture the bike, perform quality checks, and handle packing and shipping. The throughput time is measured in hours. Other criteria regarding labor cost saving are measured in terms of reduced labor hours in performing tasks multiplied by the laborer's hourly wages. Finally, reduced inventory measures the percentage of frames and components/parts that can be reduced while making the same amount of bikes, on a yearly basis.

The proof-of-concept pilot system in this study shows that the application of RFID and agent technology can improve the overall manufacturing process control and monitoring. Table II shows the estimated improvement if the proposed RFID based multiagent control system is adopted for XMbike's manufacturing plant. These estimated figures are drawn from a survey of several XMbike managers in charge of manufacturing and logistic operations. Their evaluations are based on our previous research [13], the performance of the prototype system, and company's internal statistics about plant operations and labor costs.

The proposed framework is able to cost-effectively track WIP, close the gaps between product flow and information flow, reduce inventory, and greatly enhance the performance of the company's just-in-time and just-in-sequence operations.

Current RFID system architectures require that RFID readers relay their information to a middleware located on a centralized server. Readers must also be wired to network systems, making them costly to deploy and hard to reconfigure. Unlike current centralized RFID systems, agent-based SEP devices are much easier to deploy and reconfigure because of their decentralized architecture and local intelligence, making them able to work independently of a centralized system and coordinate with other SEP devices.

C. Challenges for Future Implementation

Environmental interference with the reading of passive tags is one of the major challenges of UHF RFID technology. UHF tags are sometimes difficult to read when placed within a few millimeters of a metal surface. As a result, we had to purchase anti-metal tags and deploy more antennas to capture a tag's signal from different angles. These extra measure somewhat increase implementation costs. The tag writing operation may sometimes fail, but this problem has been greatly improved as we adopted RFID readers with higher quality and reliability for SEPs. The proposed virtual tag architecture somewhat reduces this risk.

As for agent technology, there are some technical challenges concerning the design of communication and coordination protocols. Putting an agent in an embedded device and enabling it to socialize with other devices or agents is not a trivial task. Many technical challenges did arise during SEP implementation. However, even though more challenges await as we try to extend the original prototype system, we believe that integrating RFID and agent technologies is an interesting topic that can benefit the industry in the long run.

VIII. CONCLUSION

As fierce global competition and mass customization of industrial products increase, manufacturing firms are increasingly transitioning into ubiquitous and real-time organizations. However, it is nearly impossible to become a real-time enterprise without seamlessly integrating information flows and physical object flows. The proposed RFADP framework and prototype system demonstrated in this study offers a novel approach to developing a distributed RFID system that leverages RFID and SEP technology to support the real-time integration of information and physical flows and to facilitating real-time operational control. However, the mechanisms that facilitate agent coordination are limited in this prototype system and future research should expand these mechanisms to account for more complex agent coordination schemes like bargaining and auction. The proposed solution can thus serve as a reference model for the surveyed company and many other manufacturers wishing to build next generation manufacturing information systems. In summary, this research makes the following contributions.

- This study proposes a RFID and agent-based manufacturing control and coordination framework which can better facilitate the tracking and controlling of dynamic manufacturing process flows, improve traceability and visibility of WIP, and reduce inventory.

- Based on this framework, we built a SEP device that can simplify the deployment of RFID and agent technology and maximize the potential benefits of these technologies.
- The prototype system demonstrates the successful application of UHF RFID technology in a manufacturing environment.
- The findings of this research reveal that the prototype system can provide both shop floor operators and production supervisors with real-time production process information, helping them respond to the status of the production line in real time and make better decisions in handling production events.

REFERENCES

[1] D. C. Twist, "The impact of radio frequency identification on supply chain facilities," *J. Facil. Manag.*, vol. 3, no. 3, pp. 226–239, 2005.

[2] M. Karkkainen and J. Holstrom, "Wireless product identification: Enabler for handling efficiency, customisation and information sharing," *Supply Chain Manag.: An Int. J.*, vol. 7, no. 4, pp. 242–252, 2002.

[3] B. J. Pine, II, *Mass Customization: The New Frontier in Business Competition*. Boston, MA: HBS Press, 1993.

[4] J. Wind and A. Rangaswamy, "Customerization: The next revolution in mass customization," *J. Interact. Market.*, vol. 15, no. 1, pp. 13–32, 2001.

[5] R. Kalakota, J. Stallaert, and A. B. Whinston, "Implementing real-time supply chain optimization systems," in *Proc. Conf. Supply Chain Management*, Hong Kong, Aug. 1995.

[6] Z. Stojanovic, A. Dahanayake, and H. Sol, "Modeling and design of service-oriented architecture," in *Proc. IEEE Int. Conf. Systems, Man and Cybernetics*, 2004, pp. 4147–4152.

[7] K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*. New York: Wiley, 2003.

[8] D. McFarlane, S. Sarma, J. L. Chirn, C. Y. Wong, and K. Ashton, "Auto ID systems and intelligent manufacturing control," *Eng. Appl. Artif. Intell.*, vol. 16, pp. 365–376, 2003.

[9] G. G. Meyer, K. Framling, and J. Holmstrom, "Intelligent products: A survey," *Comput. in Ind.*, vol. 60, pp. 137–148, 2009.

[10] T. A. Byrd, B. R. Lewis, and R. W. Bryan, "The leveraging influence of strategic alignment on IT investment: An empirical examination," *Inform. & Manag.*, vol. 43, no. 3, pp. 308–321, 2006.

[11] R. Hackathorn, "Minimizing action distance," *The Data Admin. Newsletter*, no. 25, 2003.

[12] M. R. Liu, Q. L. Zhang, L. M. Ni, and M. M. Tseng, "An RFID-based distributed control system for mass customization manufacturing," *Lecture Notes in Comput. Sci.*, vol. 3358, pp. 1039–1049, 2004.

[13] R.-S. Chen, Y.-S. Tsai, and A. Tu, "An RFID-based manufacturing control framework for loosely coupled distributed manufacturing system supporting mass customization," *IEICE Trans. Inform. & Syst.*, vol. E91–D, no. 12, Dec. 2008.

[14] J. Neelamkavil, W. Shen, Q. Hao, and H. Xie, "Making manufacturing changes less disruptive: Agent-driven integration," in *IFIP Int. Federation for Inform. Process.*, 2006, vol. 220, Information Technology for Balanced Manufacturing Systems, pp. 271–280.

[15] A. W. Colombo, R. Schoop, and R. Neubert, "An agent-based intelligent control platform for industrial holonic manufacturing systems," *IEEE Trans. Ind. Electron.*, vol. 53, no. 1, pp. 322–337, Feb. 2006.

[16] A. J. C. Trappery, T.-H. Lu, and L.-D. Fu, "Development of an intelligent agent system for collaborative mold production with RFID technology," *J. Comput. Integr. Manufact.*, vol. 20, no. 5, pp. 423–435, Jul. 2007.

[17] Marik, D. McFarlane, and P. Valckenaers, "Evaluating a holonic packing cell," *Lecture Notes in Artific. Intell.*, vol. 2744, pp. 246–257, 2003.

[18] C. Floerkemeier and M. Lampe, "Issues with RFID usage in ubiquitous computing applications," *Lecture Notes in Comput. Sci.*, vol. 3001, pp. 188–193, 2004.

[19] L. Zhekun, R. Gadh, and B. S. Prabhu, "Applications of RFID technology and smart parts in manufacturing," in *Proc. DETC'04: ASME 2004 Design Engineering Technical Conf. and Computers and Information in Engineering Conf.*, Oct. 2004.

[20] R. G. Qiy, "RFID-enabled automation in support of factory integration," *Robotics and Comput.-Integr. Manufact.*, vol. 23, no. 6, pp. 677–683, Dec. 2007.

[21] G. Q. Huang, Y. F. Zang, and P. Y. Jiang, "RFID-based wireless manufacturing for walking-worker assembly islands with fixed-position layouts," *Robot. and Comput.-Integr. Manufact.*, vol. 23, no. 4, pp. 469–477, 2006.

[22] R. Want, K. Fishkin, A. Gujar, and B. Harrison, "Bridging physical and virtual worlds with electronic tags," in *Proc. ACM CHI '99*, Pittsburgh, PA, 1999, pp. 15–20.

[23] K. H. Harry, K. L. C. Chow, W. B. Lee, and K. C. Lau, "Design of a RFID case-based resource management system for warehouse operations," *Expert Syst. with Applic.*, vol. 30, no. 4, 2006.

[24] D. Paret, *RFID and Contactless Smart Card Applications*. New York: Wiley, 2005.

[25] K. Romer, T. Schoch, F. Mattern, and T. Dubendorfer, "Smart identification frameworks for ubiquitous computing applications," in *Proc. IEEE Int. Conf. Pervasive Computing and Communications*, Dallas-Fort Worth, TX, 2003, pp. 253–262.

[26] S. R. Aroor and D. D. Deavours, "Evaluation of the state of passive UHF RFID: An experimental approach," *IEEE Syst. J.*, vol. 1, no. 2, pp. 168–176, Dec. 2007.

[27] S. J. Russell and P. Norvig, *Artificial Intelligence/A Modern Approach*. Englewood Cliffs, NJ: Prentice-Hall, 2003.

[28] N. R. Jennings and M. J. Wooldridge, "Applications of intelligent agents," in *Agent Technologies: Foundations, Applications, and Markets*. New York: Springer Verlag, 1998.

[29] F. R. Lin, G. W. Tan, and M. J. Shaw, "Modeling supply-chain networks by a multi-agent system," in *Proc. IEEE 31st Annu. Hawaii Int. Conf. System Science*, 1998, pp. 5–114.

[30] J. J. Jeng, J. Schiefer, and H. Chang, "An agent-based architecture for analyzing business processes of real-time enterprises," in *Proc. IEEE Conf. Enterprise Distributed Object Computing*, 2003, pp. 86–97.

[31] N. G. Odrey and G. Mejia, "A re-configurable multi-agent system architecture for error recovery in production systems," *Robot. and Comput. Integr. Manufact.*, vol. 19, no. 1-2, pp. 35–43, 2003.

[32] J. Ferber, *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Reading, MA: Addison-Wesley, 1999.

[33] The Foundation for Intelligent Physical Agents FIPA [Online]. Available: http://www.fipa.org/

[34] M. Obitko and V. Marik, "Ontologies for multi-agent systems in manufacturing domain," in *Proc. IEEE 13th Int. Workshop on Database and Expert Systems Applications (DEXA'02)*, Aix-en-Provence, France.

[35] R. Chen and D. Chen, "Apply ontology and agent technology to construct virtual observatory," *Expert Syst. Applicat.*, vol. 34, pp. 2019–2028, 2008.

[36] Java Agent Development Framework JADE, 2007 [Online]. Available: http://jade.tilab.com/

[37] K. Romer, T. Schoch, and F. Mattern, "Smart identification framework for ubiquitous computing applications," *Wireless Netw.*, vol. 10, no. 6, pp. 689–700, 2004.

[38] N. Chokshi, A. Thorne, and D. MaFarlane, "Routes for integration auto-ID systems into manufacturing control middleware environments," *White Paper, Auto-ID Center*, 2004.

[39] C. Floerkemeier, C. Roduner, and M. Lampe, "RFID application development with the ACCADA middleware platform," *IEEE Syst. J.*, vol. 1, no. 2, pp. 82–94, Dec. 2007.

[40] X. Su, C.-C. Chu, B. S. Prabhu, and R. Gadh, "On the identification device management and data capture via WinRFID1 edge-sever," *IEEE Syst. J.*, vol. 1, no. 2, pp. 95–104, Dec. 2007.

[41] D. Weyns, H. Van Dyke Parunak, F. Michel, T. Holvoet, and J. Ferber, "Environments for multiagent systems state-of-the-Art and research challenges," *Lecture Notes in Artific. Intell.*, vol. 3374, pp. 246–257, 2003.

[42] J. D. Musa and A. I. K. Okumoto, *Software Reliability-Measurement, Prediction, Application*. New York: McGraw-Hill, 1987.

Mengru (Arthur) Tu received the M.S. degree in computer science from Northwestern Polytechnic University, Fremont, CA, and the MBA in information management from the University of Texas at Austin. He is currently pursuing the Ph.D. degree at National Chiao Tung University (NCTU), Hsinchu, Taiwan.

He is a Researcher with both the Industrial Technology Research Institute (ITRI) at NCTU and the Identification and Security Technology Center of the ITRI. His research interests include radio frequency identification and real-time enterprise, multiagent systems, and artificial intelligence.

Jia-Hong Lin received the M.S. degrees in electrical engineering from National Cheng-Kung University, Tainan, Taiwan, in 2006. He is currently pursuing the Ph.D. degree at the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan.

His research interests include embedded computing controller design, multiagent systems, artificial intelligence, and multivariable control.

Ruey-Shun Chen received the Ph.D. degree from the Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu, Taiwan.

He is an Associate Professor with the Department of Information Management at China University of Technology, Hukou Township, Hsinchu. His research interests include radio frequency identification, data mining, genetic algorithms, performance evaluation, business networks, and distributed systems.

Kai-Ying Chen received the B.S., M.S., and Ph.D. degrees in mechanical engineering from National Taiwan University, Taipei, Taiwan, in 1988, 1990, and 1996, respectively.

He then served as an Engineer at the Industrial Technology Research Institute (ITRI), Hsinchu, Taiwan, for four years. He is currently an Associate Professor with the Department of Industrial Engineering and Management, National Taipei University of Technology, Taipei, Taiwan. His research interests are in the field of shop floor control, management information systems, Petri-Net modeling, and radio frequency identification (RFID) applications.

Jung-Sing Jwo received the B.S.E. degree from National Taiwan University, Taipei, Taiwan, in 1984, and the M.S.E. and Ph.D. degrees in computer science from the University of Oklahoma, Norman, in 1988 and 1991, respectively.

He is currently with the Department of Computer Science, Tunghai University, Taichung, Taiwan. He held an Overseas Professorship at the Software Technology Institute, Zhejiang University, Hangzhou, China. His research interests include distributed computing, enterprise computing, and software engineering.

Dr. Jwo is a member of the Software Engineering Association of Taiwan (SEAT).