<table>
<thead>
<tr>
<th><strong>Title</strong></th>
<th>RFID-enabled real-time manufacturing execution system for discrete manufacturing: Software design and implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Author(s)</strong></td>
<td>Zhong, RY; Huang, GQ; Dai, QY; Zhou, K; Qu, T; Hu, GJ</td>
</tr>
<tr>
<td><strong>Citation</strong></td>
<td>The 2011 IEEE International Conference on Networking, Sensing and Control (ICNSC 2011), Delft, the Netherlands, 11-13 April 2011. In Proceedings of ICNSC, 2011, p. 311-316</td>
</tr>
<tr>
<td><strong>Issued Date</strong></td>
<td>2011</td>
</tr>
<tr>
<td><strong>URL</strong></td>
<td><a href="http://hdl.handle.net/10722/135889">http://hdl.handle.net/10722/135889</a></td>
</tr>
<tr>
<td><strong>Rights</strong></td>
<td>Creative Commons: Attribution 3.0 Hong Kong License</td>
</tr>
</tbody>
</table>
Abstract—Discrete manufacturing (DM) refers to produce products in non-sequential processes so as to respond to market and customer requirements quickly under limited lead-time. However, in shop-floor management, DM companies usually confront challenges such as information gaps between different manufacturing units, slow responsiveness to customer changes, and poor visualization. The main reasons are lacking of efficient manufacturing data collection manners and software to support shop-floor management. This paper introduces an RFID-enabled real-time manufacturing execution system (RT-MES) for improving DM shop-floor management level in the perspective of illustrating the RT-MES software design and implementation. Several contributions from this paper are significant. First, a framework of RFID-enabled RT-MES is proposed, which is generic and helpful for enterprise information system (EIS) construction. Second, a plug-universal database-aided design (PUDAD) concept and its realization are elaborated, which could reduce RT-MES development and implementation cycle. Third, an interface middleware is reported to enable RT-MES real-time intercommunication with other EISs such as SAP ERP. Fourth, a real-life case study describes how RT-MES to enhance a typical DM firm’s shop-floor management, which can be referenced by other DM companies when they initiate and implement RFID-enabled solutions.

I. INTRODUCTION

ISCRETE manufacturing (DM) is characterized by individual or separated unit production and thus applied in many segments due to its quick responsiveness to market and customer requirements (www.wikipedia.org). It plays an important role in a country’s economy since most of critical manufacturing fields belong to DM, including large-scale machinery manufacturing, mold & die manufacturing and so forth. DM often features dynamic and complicated manufacturing environment. Hence it faces great many of challenges in the shop-floor management, including manual and paper-based data collection, weak ability of responsiveness and poor visualization [1].

MES (Manufacturing Execution System) is possible to address these challenges. Since MES is an advanced shop-floor management system, consisting of several functionalities such as automatic production manage, quality control, logistics tracing and other related management tools [2]. It concentrates on the management from the start of the orders to the end of the products on the manufacturing shop-floor [3]. In the shop-floor management such as quality management, dispatching manufacturing unit, and data management, MES enables to provide most primary production information on one hand. It can utilize the information for guiding manufacturing activities on the other hand [4-6].

However, most of researchers set up MES models and frameworks from the view of software modeling and emulation mode instead of consideration of practical production system and developing environment. The gaps of research studies and practical implementations negatively affect the movements of MES in DM fields given the specialty of its manufacturing complexity. Real-life studies from DM companies are scarcely reported from the literature due to some reasons. First, the complexity challenges to propose a holistic approach to deal with all the issues. Second, the manual and paper-based operation system cannot support the information system in DM manufacturing shop floors. Hence, a basic step is to facilitate them with efficient data collection facilities. Finally, MES implementation and development are new advanced technology. It is still under development in accordance with other technologies such as RFID.

This paper focuses on fulfilling some gaps that DM companies are interested when they contemplate RFID-enabled MES, e.g. how to design, develop and implement an RT-MES in a cost-efficient way, how to use some tools to support and reduce system/software development cycle, and how to integrate RT-MES with other systems.

II. OVERVIEW OF RFID-ENABLED REAL-TIME MANUFACTURING EXECUTION SYSTEM (RT-MES)

A. RT-MES Framework
RT-MES has four layers: Shop-floor Layer, MES Layer, Interface Layer and Decision-making Layer (Figure 1). Shop-floor Layer includes various hardware devices (e.g. readers, tags, communication devices etc). [7] gives detailed information about this layer in terms of RFID devices deployment, network deployment, the hardware platform and its working mechanism. MES Layer contains three core services: communication service, planning and scheduling service and visualization service. Interface Layer is an interface service that is based on middleware philosophy aiming at real-time intercommunicating with other systems. Decision-making Layer encompasses some enterprise information systems (EISs) such as ERP (Enterprise Resources Planning), PDM (Product Data Management), and CAPP (Computer-aided Process Planning), which are primary for making decisions such as planning and scheduling.

Fig. 1. RT-MES Framework

The RT-MES framework forms a closed-loop by enabling information flow from down level (e.g. frontline manufacturing sites) to the top level (e.g. SAP ERP). RT-MES uses RFID devices to collect real-time manufacturing data firstly. Secondly, the collected data are sent to RT-MES software to process, such as converting the data into information, supporting for planning and shop-floor scheduling as well as visualizing various production information for helping shop-floor management. Thirdly, the interface service sends real-time critical information into decision-making systems for further or high level strategy evaluation and selection.

B. Key Services

RT-MES software includes four vital services which play critical role in the closed-loop shop-floor management. Communication Service manages all data transferring such as RFID readers and tags, readers and workstations. Its detailed principle and working operations were illustrated in [7].

Planning and Scheduling Service mainly executes real-time shop-floor planning and scheduling. The aims of this service are to create a real-time manufacturing environment on DM shop floors, including auto-assigning tasks, real-time monitoring Work-In-Progress (WIP) items and real-time tracing and tracking logistics. This service is based on a critical concept which is real-time job pool. The real-time job pool is detailed illustrated in [8].

Visualization Service visualizes the information such as machine working status, material delivery, WIP tracking and tracing information as well as real-time workstation material requirements. This service enables shop-floor managers to real-time monitoring the material delivery due to the up-to-date consumption of material is captured by RFID devices. As a result, they can make quick responsiveness on which materials have to be delivered to where at what time. Meanwhile, logistics workers get the logistics tasks when they put their staff card. The logistics information can be transferred and visualized on time basis.

Interface Service is a bridge between RT-MES and other EISs. Its main functionality is real-time data intercommunication, aiming at enhancing information sharing. This service enables real-time information delivery from shop-floor to the high-level entities on one hand, the corresponding feedback from high-level decision-making units to low-level execution cells will be delivered timely on the other hand. In this manner, closed-loop information flow is formed so that the managerial level is strengthened within a company, especially shop-floor management.

Since the RT-MES and its key services can facilitate DM company’s production operations and management, several challenges exist when a company design and initiate this RFID-enabled solution. The first challenge is how to manage the shared resources (e.g. database) since RT-MES includes four key services which should be developed parallel due to the limited development cycle and testing strategy. The second challenge is how to realize configuration-driven development in RT-MES so as to reduce the process of writing source code. The third challenge is how to ease the development work in secondary exploitation of RT-MES. In order to tackle these challenges, this paper proposes a novel software development concept and its methodology so-called Plug-Universal database-aided design (PUDAD).

III. PLUG-UNIVERSAL DATABASE-AIDED DESIGN

Plug-Universal database-aided design (PUDAD) is a new software development manner getting from large number of practical EIS’s implementation. Its main principle is based on a 3rd party tool for aiding the software development, instead of operating resources in the database directly. The tool works as plug universal power adapter that can work at any regions according to the predefined standards.

USB disk, for example, can be plugged into the USB port and the information stored in the disk can be accessed. In the similar way, a software/procedure tool can replace the USB disk in software cases so as to help software developers to manage database sources. Additionally, the tool intends to parallel develop software and implement it through configuration-driven development so as to enhance efficiency and reduce software development cycle [9].

A. Plug-universal design (PUD)
PUD is a software development method using plug universal concept to help software engineer to develop systems through indirectly managing the some sources such as database, development platform and programming language. Its principles are illustrated in Fig. 2. The principles of PUD can be elaborated in 4 aspects from practical example to software design case. The similarities and main concepts of each aspect are explained as follows.

1: electric appliances ↔ software application (App)

Electric appliances in our daily-life are standardized by different types of switching power adapters/supplies such as GS (EU) (PAHS+REACH), BEAB (UK), BSMI (Taiwan), PSE (Japan) etc. When travelling, we expect a changeable plug adapter that can be used in different regions. Similarly, in software design, there may be many components/procedures which frequently involve the shared sources such as database. Unlike the electric appliances, these components may parallel operate to the database elements directly (e.g. a data table, a storage procedure etc). As a result, some tools are necessary to control the sources due to the large number of data.

2: plug universal adapters ↔ PUD tools

In practical cases, the changeable plug adapter can address the problem when facing the heterogeneous switching power adapters/supplies. Similarly, we propose the concept of PUD to develop some tools for the same purposes. These tools can be plugged into the SDE so as to manage different sources such as database. While Apps design, especially the programmers, operate the databases directly such as SQL server, Access, and Oracle. They are frequently facing the challenges of repeated work due to the minor modification of database sources (such as adding or deleting a column in a data table). These challenges could be solved by the tools based on PUD. For example, modification of database including data tables, storage procedures etc, will be managed by these tools first. Second, the tool will inform the corresponding components and its programmers immediately. Third, programmers can revise their involved parts specifically.

3: power socket ↔ SDE

In real-life cases, the power socket is connected directly to the electricity sources. It works as a bridge between functional electric appliances and the power. Similarly, SDE is a platform that acts as a bridge between data sources (database) and final pieces of software. Commonly, the software developers, especially the programmers, operate the databases directly such as SQL server, Access, and Oracle. They are frequent facing the challenges of repeated work due to the minor modification of database sources (such as adding or deleting a column in a data table). These challenges could be solved by the tools based on PUD. For example, modification of database including data tables, storage procedures etc, will be managed by these tools first. Second, the tool will inform the corresponding components and its programmers immediately. Third, programmers can revise their involved parts specifically.

4: Electricity ↔ database

Electricity, in real-life, plays a critical role in driving the normal functionality of electric appliances. Similarly in software design, the database acts as the “blood” throughout the whole software. However, the database sources are much more challengeable to manage than electricity due to some reasons. Firstly, the data may be parallel operated by software. Therefore, it is necessary to keep their consistence. For instance, a data table is operated by two different programs. One wants to update a record into the table. The other wants to read the data from the table. How to deal with this operation is a big challenge. Secondly, there are various sources in the database. Different sources maybe managed by different manners. For example, data tables can be accessed directly through program. Storage procedures are usually called through parameters. As a result, that increases the managerial complexity.

One solution frequently used by the development team is based on the documents named data dictionary (DD) that is centralized repository of information about databases. DD has several limitations. In the first place, DD is usually documented in the form of Word/Excel files, which are difficult to query. Moreover, they cannot be informed to the all involvers immediately if there are some modifications. Furthermore, the front-line programmers are busy with writing the source code. They might miss the modifications unless they could be informed on time. Finally, the above limitations cause “snow-ball” effects such as the high frequency of repeated writing of source code and software scope creep.

Some aspects should be considered in software development using PUDAD. First of all, the parameters should be correct while inputting to PUDAD. Otherwise it gives warnings. Secondly, when there are some changes in database, programmers only have to revise the parameters. The corresponding functional modules will be revised automatically. Thirdly, in the case of testing and secondary exploitation of software, it is followed configuration-driven development instead of writing any source codes. That results easy-going and self-defined function modules given the different user’s requirements. PUDAD reduces not only the development cycle, but also the cost.

RT-MES has been successfully developed by using PUDAD in a short time. Due to its data sharing level, it is fundamental for RT-MES to share data with other EISs. For example it is important to get data from ERP for executing the manufacturing tasks and reporting tasks progresses to CAPP so that the machine utilizations can be estimated and products cost can be calculated accurately. We propose an interface middleware for this purpose.
IV. INTERFACE MIDDLEWARE

This paper takes a seamless integration of SAP ERP for example to illustrate how to establish an interface to bridge these two systems so as to realize two categories of data intercommunication. One is real-time. Another is timed. We utilize the middleware methodology to design an interface so-named interface middleware for this aim [10].

A. Interface Middleware

Interface middleware is a middleware that runs in the Server, aiming at transferring data between SAP ERP and RT-MES. It includes two aspects: real-time and timed transmission. First, the critical data such as material consumption, order status etc are real-time sent to SAP for further planning and decision-making. In some cases, real-time transferring can be executed by manual operation when facing urgent production orders. The production orders, job instructions etc are delivered from SAP to RT-MES through timed strategy.

B. Working Mechanism

The working mechanism of interface middleware is based on SAP R3 architecture which contains three servers: central DB (Database), application server and customer server combining with the advantage of RFC (Remote Function Call) and BAPI (Business Application Programming Interfaces)[10]. RFC and BAPI interfaces of SAP are used for communication between SAP and other systems from an external program so that individual requirements are easily satisfied. There are many levels that external applications can integrate with SAP. BAPI provides a programming interface to business objects which can be manipulated by this method.

The figure 3 describes the working mechanism of interface middleware. Two queues play the critical role. First is real-time queue, which controls the real-time data that will be processed. The components of interface middleware use DLLs to process the data that mainly come from RT-MES. The DLLs are designed by SOCKET technology. Another queue is timing queue, which manages the timed data processing. For example, production orders must be transferred to RT-MES in the beginning of working day. Therefore, this queue executes this operation through pre-configured time (e.g. 4:00am) every day given the operation frequency of database.

Using the interface middleware to communicate, RT-MES and SAP have to follow some agreed protocols. Therefore, we design several SAP connectivity modules to control and process the received data from RT-MES to uniform the heterogeneous data such as data format, storage method etc. These modules are based on the concept of “mutual design” which meets the data standardization among different systems. Mutual design means using different procedures to operate the data in two systems. The systems are heterogeneous in terms of realized platform, data format and running system. As a result, different systems must open an interface to unify the data according to certain protocols.

Take data format for example, the data structure is defined as following Table II in this case. There are six parts. Start Flag uses ‘#1’ to identify. Interface ID utilizes the name of each function unit as ID due to its easy understanding. Data Length confines the maximum data size, which is ‘1024K’ in this case because of the network bandwidth. If the data size is larger than that, it will be divided into several units. Data Contents is a character string. Transfer Type is an integer. ‘1’ means timed transfer scheme. ‘0’ means real-time scheme. ‘#127’ represents the End Flag.

<table>
<thead>
<tr>
<th>Start Flag</th>
<th>Interface ID</th>
<th>Data Length</th>
<th>Data Contents</th>
<th>Transfer Type</th>
<th>End Flag</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>OrderRelease</td>
<td>1024K</td>
<td>&quot;*****&quot;</td>
<td>1</td>
<td>#127</td>
</tr>
</tbody>
</table>

V. CASE STUDY

The case company is Guangdong Keda Electrical and Mechanical Co., Ltd., (Keda for short). Keda was founded in 1992, which is located in the private economic prosperity Guangdong province. The company is a typical discrete manufacturing enterprise and specially to manufacture the ceramic, stone, wall material, energy saving and environmental protection, and other large-scale integration of mechanical equipment. It currently employs more than 2,000 workers. Keda is at the forefront of adopting EISs to facilitate its operations and management. Since from 2000, it has been equipped with SAP ERP, CAPP, PDM, CAD and OA.

Despite all the efforts, there are still some problems due to the complex situations on the shop floors that the above EISs are hardly to cope with. In the first place, machine utilization and personnel measurement are ambiguous since it lacks of shop-floor management system to monitor the manufacturing sites. Moreover, data on shop floors are managed by manual operations and paper-based system. Therefore, Keda is forced to spend lots of time and efforts to address large number of paper-based documents and sheets when querying and counting. Furthermore, logistics and WIP (Work-In-Progress) items are difficult to control due to the usage of paper-based identification cards on shop floors. The cards are always missing and damaged. Finally, frequent engineering changes
and customer variability disturb shop-floor productions. The shop-floor management system cannot handle the disturbances [8, 11]. As a result, the potential of shop-floor production in Keda is confined.

In order to tackle these problems, an RFID-enabled real-time manufacturing execution system (RT-MES) is implemented on its shop floors. RT-MES facilitates and improve the company’s shop-floor management in several aspects. Firstly, RT-MES uses RFID technology to identify its manufacturing objects so as to capture real-time data. Therefore, real-time manufacturing shop-floor is created. Secondly, the planning and scheduling service in RT-MES uses the RFID data and upgrades the production into real-time level. The real-time working mechanism is illustrated in the following steps.

1. Production orders from SAP, job instructions from CAPP and material requirement plans from PDM are transferred to planning and scheduling service through interface;
2. Planning and scheduling service uses real-time job pools to deal with the job distribution on the shop floors. Job distribution follows hybrid flow-shop scheduling system which contains several stages. Each stage is equipped by machines with similar function.
3. Each machine is equipped by a RFID reader. The reader enables machine operators to get jobs from the job pool through reading their staff cards. Machine operators pick up a tag which is attached on the materials before starting the operations. That informs the actual start of processing work.
(4) Machine operators can also input data through the readers, including machine status, order fulfillment etc. These data are real-time feedback to RT-MES, and then sent to SAP and other EISs through interface middleware.

Thirdly, the development cycle of RT-MES is greatly reduced by using PUDAD methodology. Generally, it takes more than half a year to fulfill the project since its scope and lack of workforce. However, it mainly takes 3 months to accomplish design and programming of RT-MES software, only three software engineers mainly involved. There are several successful factors. The first is the assistance of PUDAD. PUDAD enables configuration-driven development which can reduce source code programming. This development method uses parameters for designing various functional modules. The second is the effective and efficient management of database resources. PUDAD helps the software engineers to manage data tables, storage procedures.

Fig. 4. Comparison of Before and After
and so forth through real-time visualizing their columns, real-time tracking their revisions as well as informing the programmers. The third is collaborative work with technicians and employees in the case company. This collaboration not only promotes the project progress, but also elevates the team communication level. Therefore, this spirit will be continued in their future co-operations.

Fourthly, interface service utilizes middleware philosophy to communicate with other EISs. The service enhances the information sharing level and data transmission between heterogeneous systems. That forms an information closed-loop in the case company. That means the frontline manufacturing data collected by RFID devices can reach the decision-making utilities timely, while the decisions can be got by the frontline sites as well. The closed-loop further improves Keda’s shop-floor management in terms of real-time data collection, reduction of WIP (work-in-progress) inventory and reduction of manufacturing cycle. Figure 4 gives some improvements through comparing before and after.

VI. CONCLUDING REMARKS

This paper discusses a RFID-enabled Real-time Manufacturing Execution System (RT-MES) for DM company in the perspective of software design and implementation. The RT-MES framework is proposed in terms of its layers and key services firstly. Moreover, Plug-universal database-aided design (PUDAD) is reported on how to realize configuration-driven development. Furthermore, an interface service based on middleware concept is elaborated to real-time intercommunicate with other EISs. Finally, a case study from a real-time DM company is illustrated to report how RT-MES improve the shop-floor management.

From the real-life case study, lessons are obtained from the RFID-enabled solution design and implementation. The RFID-enabled RT-MES can improve the DM shop-floor management in terms of real-time data collection and intercommunication, real-time planning and scheduling as well as efficient tracing and tracking of WIP items. During the RT-MES design and development, PUDAD methodology achieve configuration-driven development which reduce the time and cost. Additionally, interface middleware intended to improve the information sharing within heterogeneous systems is significant due to the large number of EISs. Insights and lesions could be referenced by academics and practitioners when they study or implement RFID-enabled software in other DM companies which are confronting the similar challenges in shop-floor management.

Further research and improvement are also necessary if this RT-MES is of great merits to promote. Firstly, real-time hybrid scheduling model will be investigated since most of the off-shelf models do not consider RFID-enabled real-time data in shop-floor scheduling. That mainly because RFID-enabled implementations are scarcely reported.

Secondly, emerging technology such as Data Mining (DM) could explore the historic RFID data so as to get more precise scheduling through providing corresponding parameters such as set-up time, processing time and logistics information. Therefore, it is significant in the future when RFID technology is widely used due to the new network-Internet of Things (IoT) which was proposed by Foundation of Auto-ID center of MIT (Massachusetts Institute of Technology) in 1999. The basic of IoT is RFID technology. Manufacturing fields is the starting of various things. Therefore, RFID implementation in this region is significant in the future.

ACKNOWLEDGMENT

Authors would like to acknowledge technical supports from various institutions and their technicians. The authors give special acknowledgments to Guangdong Keda Electrical and Mechanical Co., Ltd., as well.

REFERENCES