An Extensible Event-Driven Manufacturing Management with Complex Event Processing Approach

Y.H. Zhang[1], Q.Y. Dai[1*], and R.Y. Zhong[2]

[1]*Faculty of Information Engineering, Guangdong University of Technology, Guangzhou, China*
daiqy@gdut.edu.cn[*], zyh5000@hotmail.com
[2] *Department of Industrial and Manufacturing Systems Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong*
zhongzry@hku.hk

Abstract

Events are crammed with numerous basic elements in today's management and manufacturing system. Especially, for a large-scale system, there are thousands of independent messages (events) happening every second. How to integrate the information and how to deal with such massive events are faced by most of the system. Event monitoring and processing has been posed in manufacturing corporations to tackle this challenge. However, well-prepared methodology and processing are lack for this challenge. In this article, we propose an extensible event-driven manufacturing platform for handling those massive events. First, holistic architecture of the platform is depicted. Next, the principle of primitive and complex event is detailed. Eventually, Complex Event Processing (CEP) approach is applied to the platform.

Keywords: *Extensible Event-driven Manufacturing Platform, Complex Event Processing (CEP), Primitive and Complex Event*

1. Introduction

Event is regarded as the meaningful change in the system [1]. Events occur in the manufacturing management system day-to-day. Customers make an order, workers receive an order, material is delivered, machines process material, and order completes. The events above all belong to this field. These events notify the system that some status has changed and expect the system to give a response.

Today the manufacturing system is large-scaled. AMR (Advanced Manufacturing Research) introduced a concept of manufacturing integrated model which consisted of 3 levels. Different level manages different types of events. At the top level, e.g. MRPII, or ERP, the hierarchy mainly handles the events about business, projects and resources. Thereby it needs the statistical, slim data which the management can use to make decisions. At the bottom level, i.e. shop floor level, the real-time, independent and numerous event messages exist and roughly depict the plant working. The middle level, i.e. MES (Manufacturing Execution System) which is connected by the top level and the shop floor level, controls and monitors the events originating from the other two levels and is seen as the communication of them. However the actual MES don't work well in the today manufacturing operation system: the business messages can't effectively transmit to the shop floor level without delay; while the plant's statistical messages can't submit to the management appropriately. The management of events in different levels is unbalanced.

In the modern manufacturing system the event sources are diverse, such as the RFID readers, the manual reports, the keyboard input and so on. In every second, thousands of events happen in the system. At the same time, the manufacturing corporations only have the limited hardware to handle all events. Anther problem has been raised: how to use the limited resource to handle the unlimited events.

In order to address the former problem, we propose an extensible event-driven platform based on the MES to handle events. The platform can: (1). connect the different parts of the system; (2). manage and monitor the event flows across all levels efficiently; (3). bridge the gap between the business level and shop floor level; (4). give customized multi-views of the system.

We build the platform on the Event-Driven Architecture (EDA) which can design the platform with modeling components loose coupling. The components (modules) in the system are based on functions. The combination of components contributes much to be flexible and easy extensible for handling event flows.

Among the components in the platform we envision a component named EBZ (Event Buffer Zone). It can confine a buffer space for event flows. And EBZ can store candidate events at the certain thresholds, and properly forwards them to the event queue through which they can be dispatched. In this way it can cope with the asymmetry between hardware resource and event flows.

In order to tackle events efficiently and feasibly, CEP (Complex Event Processing) will be applied to the platform. This approach detects the low-level primitive events, matches the predefined pattern, and creates the high-level complex events. With the help of CEP approach, the platform can manage event flows in a reasonable way.

This article first introduces the architecture of the platform, and details the constructing modules. And then definitions of primitive and complex event are given in section 4. Next we present the CEP approach model. At last, future work is proposed for the coming research to conclude this paper.

2. Literature review

Related research is categorized according to the list:

1) the module-designed of the software application
2) the method of event definition
3) the principle of CEP

The MES concept was first introduced in 1990 by AMR [9]. It explained the model of concept into 3 levels, the planning level, the executing level and the control level. And then many corporations have focused on the implementation of the MES.

The design of architecture is the main point of the implementation. [7] presents the IBM architecture of event-driven system. And it is much concerned about the active middleware of the IBM application, and it can't be able to apply to general use. [8] puts weigh on the integration of the different levels to handle event. The actual programming is an obvious problem in the manufacturing domain. [2, 8] offer the idea of the publish/subscribe infrastructure and Service-Oriented Architecture (SOA) mechanism. However, it lacks the actual example for the infrastructure.

Event is the element for the system. So the definition of an event is helpful not only for optimizing system, but also for programming. [3, 4, 7] talk about the event definitions, and give the characteristics of the different types of event. But data flow related with event has not waked so far. And how to detect them and utilize them is regarded significant.

David C. Luckham have been introduced the CEP approach in [5, 6]. These papers have presented the elaborate principles of the CEP approach. And there are lots of the theories that discuss about the CEP approach in experiments. But for the practical implementation these principles have to been adapted, and in this way the CEP can be used in the project in reason.

3. Extensible event-driven manufacturing system

Our goal of the proposed platform is to promise a truly extensible, real-time, flexible, event-driven, resource-saved architecture for event processing. The architecture will provide the connectivity between different levels, fluent communication across all the levels, management of different type of events and multi-view of the system.

The platform which is built on MES is tightly connected with the other two levels. In the traditional manufacturing system, the effective interacting action is not common in event communication. The shop floor level produces the basic working events; whereas the business level may need the statistical data. The events in different levels can not properly inform each other across levels. The platform will solve this problem.

The business decisions sometime are determined depending on one or more events. So the platform will manage not only the primitive events, but also the complex events. And the platform founds an event abstraction level above the basic events. We apply the CEP approach to the proposed platform in order to make the system to handle different types of events.

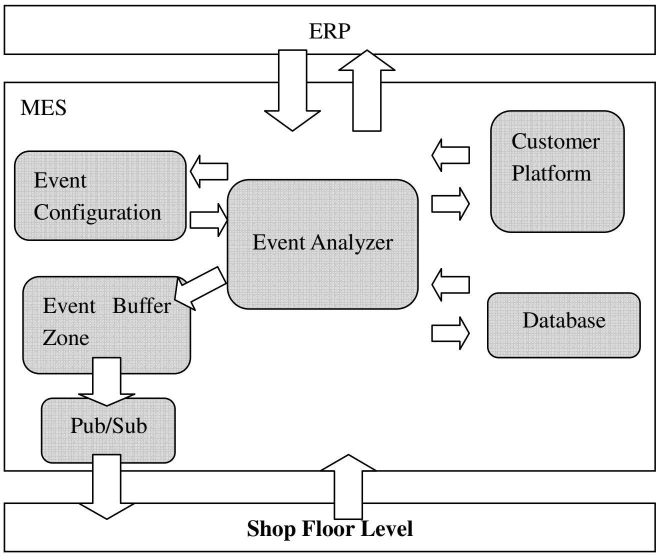


Figure 1. Architecture of the platform

The event buffer zone component is integrated in the platform. The event buffer zone component is to make a temporary room for the candidate event, and upload some of them according to some certain situation. So the platform could handle the ever-increasing events, and events won't leak out. Finally, information will not be lost.

Users of the system may have different interests in different types of events. For example, the maintain team could care about the machine fault report; the business team may be concerned about the progress of projects; the workers may take the order condition into account. And the platform uses the human-oriented UIs (user interfaces) to create the multi-view of the system, and users can focus on what they are interested in.

Figure 1 illustrates the overall architecture of the extensible event-driven manufacturing platform which comprises several components.

The event configuration is used to define and register complex events. Event analyzer is a tool to process events, no matter where events produce. The event buffer zone is the temporary zone for storing the candidate event. With the help of publish/subscribe paradigm, the platform can distribute various events. Database is necessary for storing the information of events. The UIs component can give an interface between customers and system.

3.1 Database

Database is a persistent component which can store mostly the necessary information of events which are the basic element of the system. Although the amount of information is numerous, they need to be recorded for it is essential to trace back to, analyze and process events. As well as events, the database also contains the information of identification for individuals, e.g. workers, machines, orders, etc. The database can store every individual's ID, state and authority which can enable it to start events.

Using database can be a safe way to store information. The database information can be safe in case that the power breaks down, or the computer restarts.

3.2 Publish/subscribe mechanism

Publish/subscribe (pub/sub) mechanism is an efficient way to distribute events. It doesn't connect the event producer and subscriber directly. It plays a role as an event broker. The producer only put the address-embedded events into the pub/sub, and according to the address and requirement of subscriber which is registered in the database, the pub/sub paradigm is able to forward events to the interested subscribers.

In the traditional model, the pub/sub system has an obvious drawback: the limited storing space to deal with the ever-increasing unlimited events. So in many other pub/sub systems the over-threshold events maybe leak out and some information which are brought by the events may be lost. With the help of EBZ (Event Buffer Zone), our proposed pub/sub paradigm can distribute events convincingly. The principle of EBZ will be introduced later.

3.3 User Interfaces

This component is used to found a channel on which users can communicate with the system. The UIs can give diverse views about order definition, rate of order progress, material delivery, and so on. People can choose their own views depending on their interests. With the help of CEP, the UIs will redefine your interesting novel views of the system, and the views can be customized and re-configured.

3.4 Event configuration

Event configuration is the major component of CEP approach which serves for event's register and definition. Properly speaking, it gives patterns for the event detection and corresponding reference of superior complex events. Through the UIs, the customers can define events in the system and the event definition and registration will be created in the event configuration component. When event flows are detected inside the platform, the event analyzer will first search the event register in the event configuration component. And then the event configuration will check if there is an index pointing in the event definition. Next, it will again check if the event will be matched the pattern. Finally it will give a clue to the event analyzer that which superior complex event will be created.

The advantage of the event configuration will contribute to the separation of the CEP utilities. The relation between event configuration and event analyzer is like the instruction and the execution. Because they both constitute the CEP approach, the separation of these two modules will be flexible for the system, and the respective regulation of these two modules won't influence each other.

3.5 Event buffer zone (EBZ)

Events are produced in the system every second. How to process them and not to make such huge information lost is an actual problem.

The traditional way to distribute events is to forward them into the event queue (pub/sub) directly. But the queue is a limited component. And when the coming information gets over the threshold, some of them have to be lost. The event buffer zone (EBZ) is the best choice.

EBZ is a method to create a buffer room in memory where a certain part of the events can be added. The event configuration first defines the candidate events which will be about to process. EBZ will then query the event configuration and transfer some part of the candidate events into the room. The events inside the room are all available to join in the queue, whereas the ones outside will wait for the next time to join. In this way, the number of the processing events will be restricted by the volume of EBZ. The volume also can be reconfigured by the administrator. There are three methods to load events: time-control, manual and event-driven. The time-control method is to set a period in which the EBZ will load the events; the manual way is to transfer the event if users require; the event-driven methodology is based on triggering by some events.

3.6 Event analyzer

It is the most significant component of the system. All the event flows will come through this module. The module will convert the primitive event into the complex one, connect the different part of the platform, and as a main component in the MES, give different event interfaces to the other two levels.

The event analyzer is truly the hub of the overall system. It tightly connects the top level and the bottom level, and provides the interfaces for them. The interfaces are to transform signal into standardized events, so that the events from two levels could be processed in a uniform way. The interface's discussion is beyond the scope of this article.

As the central part of the platform, it has to integrate with other components properly. First of all, it makes the information communication fluently between different components; second, with the event configuration it will process the registered events, detect them with pattern and give a correct responding action as a reply; third, it gives the reference information of the system by reading the database component; forth, through the UIs, it provides the required data to customers.

4. CEP application

Manufacturing system has become information-based in today's global competitive manufacturing market. The communication of the messages inside the system is in the digital matter. Machines, workers, orders, all these individuals have been tagged by the system. So individuals could be seen as sources of events triggered by them. The sources are both event producers and receivers. They create events, at the same time they require responding action. Various sources result in various types of events. Events are the basic element in the system. As a large-scale modern system, how to intelligently, efficiently deal with events is extremely important.

CEP is a method to enable systems to conquer this problem. CEP approach is to detect not only events, but also the relationship of events. It will aggregate several events and create a superior event to reply the events. Moreover it can filter various events and extract the meaningful ones amongst them. It is an effective method to cope with a great deal of events.

CEP has been applied largely in the business and military fields, but has not been waked in manufacturing industry.

In order to utilize the CEP approach, it has to formalize events. The event in the unified form 1) is easy to control and process; 2) can define the new event according to the standard. Currently there are several methods to define events. Some of them are to define events in a template [7]; apart from template, some of them use rule language [7, 4]. The integration of template and rule language is introduced in CEP approach in this article.

In this article, we intend to apply CEP approach to manufacturing system. In this chapter, we first introduce different types of events in the system. Next, we propose the definition of events. And then we present the principle of the CEP approach. At last we provide a case study in our platform.

4.1 Event in the system

Event is happening in the system every second. Machines report failure, material is not available, workers are much busy and so on. The events above can trigger "the_Order_Can't_Work_On" event. But in order to manage events intelligently, it is necessary for the system to immediately detect the actual reason for break down and to tell the management in time.

Based on our goal, we have to classify events into two categories, primitive event and complex event.

Primitive event: it is the basic part of the system. It only specifies the condition changing about the shop floor. Usually it contains the timestamp, the common digital values and so on. Many of the primitive events are isolated and without particular relationship with others. Therefore the primitive event can indicate the basic action of the system, but can't suggest the situation around it.

Complex event: The complex event consists of event with temporal or causal relationship. And it depends on one or more several events. When system defines a complex event, it defines not only the event itself, but also the relationship with other events. Through analyzing the complex event, the situation around it can be told. Therefore complex event constructs the abstract level of the system. It can transform the basic meaning into high-level one and inform readers that the statistical data or the subsequent action. For example, when the products have been packed, they usually could be delivered in 10 minutes, however the deliver men don't get the products in 15 minutes (the predefined condition), and the system will set an alarm (complex event) that something must make a mistake during the course between packing and delivery.

The separation of the event can make the event management easily to control. And another benefit is that this separation is preparation for the CEP approach.

4.2 Approach of the event definition

First, we have to set up an approach for events which can be a standard for the event definition and detection.

There are several approaches of definition for an event currently. For example, the IBM's MES model defines each event in templates. Each event must contain situations and conditions. The situations include the aggregation, such as 'add', 'not', 'abstract' and others [7]. The situation in the template is used to aggregate the event; the event conditions in the template utilize parameters to record, define, compare, and calculate. The advantage of template is that there is a standardized template for define each event, and the architecture of the system will be compact and easy to monitor. But the drawback of this approach is obvious. Because it uses a simple situation definition (aggregating condition) so that for a complicated manufacturing system the method of using template is not enough to explain complex events comprehensively.

In order to address this problem, we envision an approach using template and rule language together. The template of the event definition gives a standard to define each event and make the system easily to handle all events; the rule language provides the logic amongst events. In this way it is standardized and flexible to process events.

The template of the event arranges events which consist of two kinds of parameters, one is index parameter and the other is extensible parameter.

The index parameter is like a pointer. It is used to specify the temporal or causal relationship. Each event can contain one or more index parameters which can together determine the creation of superior complex event. When detecting the input events, the index parameters will be used to search for the index position in the events configuration. If the event definition and register exist, the event configuration will send the event analyze a clue that which complex output event will be produced. Usually the index parameters will be described by the order ID, product ID and so on.

Extensible parameter records information that the event should bring about, e.g. the timestamp, the executants, the machine and so on. Like the index parameter, each event can contain one or more extensible parameters. Although these parameters don't indicate the creation of the superior complex events just like index parameters, they can confine the occurrence of the complex events by some conditions which have been included in the rule language. For example, if the ratio of fault products (an extensible parameter) to good ones is below 80%, the progress will be terminated. The system can detect the event extensible parameters and rule language to make the manufacturing break.

Rule language is a method to construct the event relationship in the event-driven system. It gives the logic and reasonable relationship to the independent events. Moreover it is flexible to describe the relation around the events. There are several rule languages as the role of depicting event relationship. The most famous one is XML-rule language, i.e. Xquery and Xconcept [4]. But they are mostly used in the website application. In our case, we propose the platform on the C/S model, so we choose the SQL-rule language. The SQL-rule language has been developed for a long time and it is qualified to describe the relationship of events. It's also common in the database application. In other words, it can be incorporated in the database fluently.

4.3 Model in the CEP application

When the system is working, the event flows will go through the event analyzer. Based on the event index parameters the event analyzer looks for the coming event's definition which exist in the event configuration. Next, by analyzing the SQL-rule language, the event configuration will give a clue of creation of subsequent complex event to the event analyzer. Eventually, a complex event is created. The event analyzer is an event executant, at the same time the event configuration is an event reference.

CEP approach in our platform plays a dominating role in event abstraction and complex event creation. We present the model to simulate the CEP approach working.

The principle of CEP working has been illustrated in the Figure 2. Two kinds of element constitute the CEP approach, one is the filter and the other is the map.

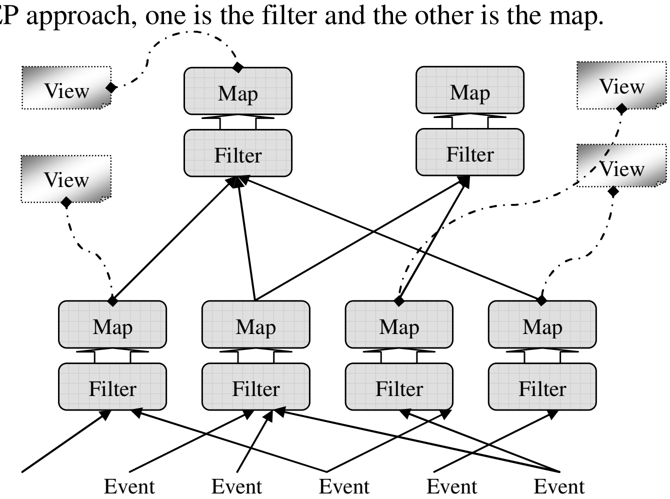


Figure 2. Principle of CEP approach

Filter is used to reduce the number of events. If the input events could go through the filter, they must be matched by patterns (conditions). While the system will select the matching events and let it go through the filter, the unmatched ones will stay outside the filter waiting for the next abstraction.

Map is used to create a complex event. When the matched events go through the filter, they will meet the map. Map will be supposed to create the superior complex events which have been determined in predefinition. The output event can be in many different forms. For example, it is a complex event that the coming event will be bound with some new parameter, or the input event's parameter will be changed, or moreover a new event will be triggered. The complex event will be processed in the recursive way, in other words the output event just like the original ones will meet another filter and map.

4.4 Multi-view definitions

Using the CEP approach, we can see the system from the different views which are illustrated in Figure 2.

Different people have interests in different events. From Figure 2, the filer abstracts the incoming events, and the map outputs the various interesting events for customers. Therefore, by filter and map, the customers can view what they are interested in from the system. In other words, the system can be displayed in different angles. People can observe the system from the multi-view according to their interests.

At the same time, the platform can not only display the predefined views of the system, but also subscribe the views. The traditional way to define the views is to program them before the system's implementation; however, the proposed platform is different: There is a human-oriented UI in the UIs components which can serve the people who know little about programming but can define the views in the easy-going UIs, so that every one can have his own view of the system. In this way the multi-view of the architecture of the system is flexible.

4.5 Case study

In this section, we introduce an actual example of CEP approach applying in our project.

Our project is based on a manufacturing corporation whose product is machine's component. So the course of product order is common in this system. The example concerning the order flowing is illustrated in Figure 3.

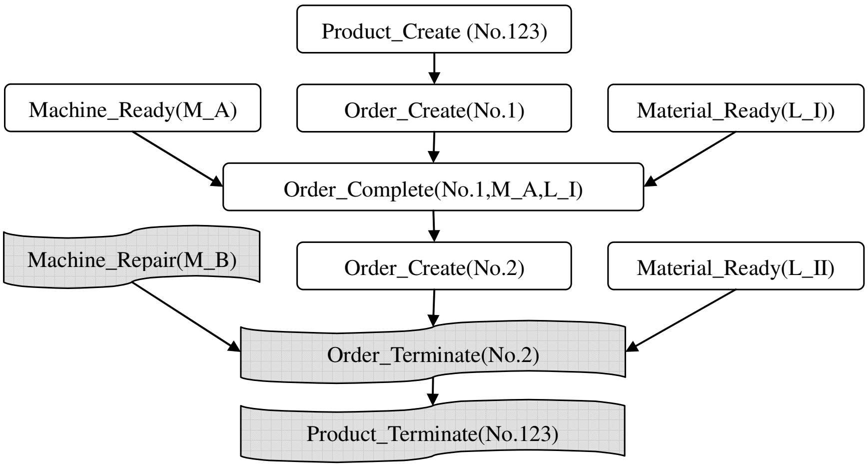


Figure 3. Course of order flow

We can see from Figure 3 that when the No.123 product is required, it needs a series orders to complete it, order No.1, order No.2, etc. Each order also needs the requirement that the machine and material both are available. When either of them is not ready, the order has to stop and at the same time the product processing will be terminated.

5. Conclusion

The application of an extensible event-driven platform with CEP approach is introduced in this paper. The platform is organized by different modules and the loose coupling between modules. The event flow can be operated and controlled in order. The platform integrates the pub/sub infrastructure and event buffer zone, in this way to propose a noble method for processing event.

The CEP approach is applied in this project. The principle of the CEP approach has been researched for many years. And it is mainly implemented in the business and military fields. This paper introduces an application of innovative design of CEP in manufacturing system, especially on MES.

With the help of the CEP approach, the system can reduce the complexity of the system, and provide the multi-view for the system.

6. Future work

Our proposed platform has contributed much to the implementation of the manufacturing systems, while it also raises different challenges for the future work. First, the pub/sub paradigm and EBZ have given an effective method to cope with various events, so the information of the event flows won't leak out. But when the scale of corporation is developing, the number of events becomes uncontrolled and the latency time between the event detection and reaction will become significant, so the bottleneck appears. The solution may be adding the number of the event analyzer and EBZ. How to cooperate with all the event analyzers and EBZs is the main point of future work. Second, the definition of event in the paper utilizes two tools, i.e. template and rule language. The template offers a standard model to define events; whereas the rule language can provide a convenient way to describe events relationship. In the practical project, the rule language is free, easy and facilitating to event definition; but in other words, the rule language lacks a unified model for illustrating events, it is difficult to adapt the original definition in the project, moreover to apply the similar definition of events to other fields. As a further topic, it is important to put weigh to develop the template definition to realize that the template may contain the more information including event relationship, in order to give a standardized, reused definition of event. Third, the model of the CEP approach must be reinforced. CEP is more discussed in theory, less introduced in the implementation fields, especially the manufacturing field. So the principle and the model optimization is the another direction of research.

Acknowledgement

The authors thank to the nation 863 (No. 2006AA04A124) team which gave partial finance support and the chance to accomplish the research. Thanks to all members of project team, Faculty of Information Engineering of University of Guangdong University of Technology, Department of Industry and Manufacturing System Engineering, HKU. Finally thank you very much to KEDA INDUSTRIAL CO., LTD.

References

[1] DAVID L., "The power of events: an introduction to complex event processing in distributed enterprise systems", Boston, Addison Wesley, 2002.

[2] B Kowalewski, M Bubak, B Balis, "An Event-Based Approach to Reducing Coupling in Large-Scale Applications", Lecture Notes in Computer Science– Springer, Volume 5103, 2008, 358-367

[3] ANTONY G, JUAN C A., "Two approaches to event definition", Database and Expert Systems Applications (DEXA), Lecture notes in computer science - Springer, Volume 2453, 2002, 705-714.

[4] F Bry, M Eckert, "A High-Level Query Language for Events", IEEE Services Computing Workshops, SCW'06, 2006, 31-38.

[5] David C.Luckham , Brian Frasca Program Analysis and Verification Group Computer Systems Lab Stanford University, "Complex Event Processing in Distributed System", 1998.

[6] J Vera, L Perrochon, DC Luckham, "Event-Based Execution Architectures for Dynamic Software Systems", Software architecture: TC2 first Working IFIP, Vol. 12, 1999.

[7] Y Magid, A Adi, M Barnea, D Botzer, E, "Application Generation Framework for Real-Time Complex Event Processing", Computer Software and Applications, COMPSAC '08, 32nd Annual IEEE International, 2008, 1162-1167.

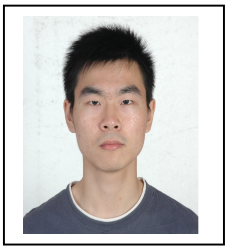
[8] K Walzer, J Rode, D Wunsch, M Groch, "Event-Driven Manufacturing: Unified Management of Primitive and Complex Events for Manufacturing Monitoring and Control", Workshop on Factory Communication Systems, 2008.
[9] Bill Swanton, "MES Five Years Later: Prelude To Phase III", www.amrresearch.com, 1995.
[10] Run-yang Zhong, Qing-yun Dai, Ke Zhou, Xin-bo Dai, "Design and Implementation of DMES Based on RFID", International Conference on Anti-counterfeiting, Security, and Identification (2008 ASID), IEEE, 475-477.
[11] Run-yang zhong, Qingyun Dai, Ke Zhou, "Realization of Program based on Plug-universal database-aided Design", 2008 International conference on Informaiton Management, Innovation Management and Industrial Engineering, IEEE, 2008, 377-380.
[12] Q.Y.Dai, R.Y.Zhong, M.L.Wang, X.D.Liu, and Q.Liu, "RFID-enable Real-time Multi-experiment Training Center Management System", International Journal of Advanced Science and Technology, 2009, Vol. 7, 27-48.

Authors

Professor Q.Y. Dai

BEng, Mphil, PHD

Professor Dai is a leading scholar of Information Systems in Guangdong University of Technology. She has been mainly supported by national R&D, GDSTD and industrial sponsors. She has published two reference books. In addition, she has published over 20 technical papers, some of which are index by SCI, EI and ISTP. Her research interest includes the application of RFID in manufacturing industry, image processing and pattern recognition, and manufacturing information technology.



Mr. Yi.H.Zhang
Master

Mr. Zhang is a master student in Guangdong University of Technology, major in Signal & Information Processing. He has took part in many a project in Guangdong province in China most of which are based on RFID technologies. His interests are manufacturing model design and manufacturing monitor.

Dr. Run.Y. Zhong

BSc, Mphil, TA, PHD

Dr. Zhong is a Phd student at the University of Hong Kong specialized in industry and manufacturing system. He has participated in several manufacturing information system application in PRD. He has published over 10 technical papers, some of which are index by EI and ISTP. His research interest contains manufacturing system, RFID application in manufacturing, and project management.