Intelligent Process Control System with RFID Cuboid

Kyunglag Kwon
Department of Computer and Information Science,
Korea University
Seochang-ri, Jochiwon-eup, Yeongi-gun,
Chungcheongnam-do, Republic of Korea
+82-41-860-1342
helpnara@korea.ac.kr

Jaejeong Chung
Department of Computer and Information Science,
Korea University
Seochang-ri, Jochiwon-eup, Yeongi-gun,
Chungcheongnam-do, Republic of Korea
+82-41-860-1342
chung@korea.ac.kr

Jaehwan Ryu
Department of Computer and Information Science,
Korea University
Seochang-ri, Jochiwon-eup, Yeongi-gun,
Chungcheongnam-do, Republic of Korea
+82-41-860-1342
berserkjan@korea.ac.kr

ABSTRACT
In recent years, there have been many attempts to connect the latest RFID (Radio Frequency Identification) technology with EIS (Enterprise Information System) and utilize them.

However, in most cases the focus is only on the simultaneous multiple reading capability of the RFID technology neglecting the management of massive data created from the reader. As a result, it is difficult to obtain time-related information such as flow prediction and analysis in process control.

In this paper, we suggest a new method called 'procedure tree', an enhanced and complementary version of PathTree which is one of RFID data mining techniques, to manage massive RFID data sets effectively and to perform a real-time process control efficiently.

We will evaluate efficiency of the proposed system after applying real-time process management system connected with the RFID-based EIS. Through the suggested method, we are able to perform such tasks as prediction or tracking of process flow for real-time process control and inventory management efficiently which the existing RFID-based production system could not have done.

Categories and Subject Descriptors
C.3 [Special-purpose and Application-Based Systems]: Process Control Systems.

General Terms
Management, Economics.

Keywords
RFID Data Mining, Process Control, Procedure Tree, RFID Cuboid

1. INTRODUCTION
Most enterprises use EIS systems such as ERP (Enterprise Resource Planning), SCM (Supply Chain Management) and e-commerce to manage business resources. In recent years, there have been many attempts by small and medium industries to utilize RFID technology, not the existing technologies such as barcode, labeling and etc. Compared with others, using RFID technology can not only save an enormous amount of labor forces, which means no human hands to scan the items one by one [1], but also guarantee high accuracy and speediness of reading rate. Research papers have been reported on efficient management of goods using benefits of RFID system as untouchability, convenience and capacity to store data [2].

However, we are unable to obtain such time-related information as prediction of process flow or analysis since the system is focused only on counting multiple items simultaneously and the massive raw data do not fit into real-time process management.

For this reason we suggest a method more appropriate to use in the enterprises for effective real-time process control using RFID data mining technique. In order to fully utilize real-time characteristics in the existing system, we need to refine and reprocess the raw data obtained from RFID into a usable form. For example, if we want to get more valuable information such as reports on the total stock in each process stage or the full path of goods we need some additional complex processes such as cleansing, filtering and aggregating the raw data by grouping specific procedure or functional units. However, we need to solve following problems to obtain information about the real-time process control.

• Problem of overhead occurring in the server caused by creation of a lot of datasets due to queries in the database.
• Difficult to retrieve human-legible information from only the raw data.
• The existing system itself is not capable of inferring the next sequence of the process or whether the current process is going properly or not.
• The system is not capable of offering relevant context-aware information at the specific point of time since it is difficult
to make exact flow analysis about current situation without human judgment.

To overcome such difficulties, we utilize “PathTree in a RFID-Cuboid,” one of RFID data mining technology that consists of three tables: (a) Info (b) Stay and (c) Map table \[3, 4\]. In case of using PathTree, we can detect not only static information at a specific point, but also the flow information about those data. Now that we can manage the dynamic information data, we can easily work out the four of the above mentioned problems efficiently. Finally, we connect the suggested method with the RFID-based EIS. After applying and implementing real-time process control management system, we then evaluate if the proposed system is more efficient and effective one compared with the existing method. We can see that the suggested system is more capable of carrying out real-time process control effectively which the existing system can hardly perform \[2\].

This paper is organized as follows. The section two provides background knowledge and techniques. Section three presents a comprehensive review of the current process control and cases. The section four outlines the architecture of our system which utilizes our suggested methodology. Section five presents an implementation and an evaluation of our system. Finally, we suggest some future research directions with our concluding remarks in section six.

2. BACKGROUNDS

2.1 RFID System and RFID Data

In the past few years, active researches and articles on development of RFID systems in a variety of applications such as trajectories of moving objects, SCM systems, access cards and highway surveillance systems have been reported. In these researches, benefits of RFID systems are stated as follows:

RFID system, unlike from barcodes, can recognize massive data wirelessly. Not having to read each item one by one, industries can save time and labor force and thus can increase productivity. Since the RFID system can store additional information in the tag itself according to the user need, retrieving any minimum meaningful information about items based on the tag data is possible. RFID system is consisted of RFID tag, reader, data base and application program. RFID data used in the RFID system has following characteristics.

- Simple data: The data read from readers located at different positions. It is composed of an ordered set of several sources (EPC, location, time), where: EPC is an Electronic Product Code which identifies an item; location is the place where the RFID reader scans the item; and time represents the moment when the reading took place \[4\].
- Large in-flood: Massive data sets are being produced because tag data are read regularly through readers. Thus tagging every item in such a large conglomerate as Wal-Mart means an enormous amount of data calling for the periodic management of massive data being produced.
- Inaccuracy: The read rate varies according to the environment of the work place. The real recognition rate of RFID is often in the 60-70% range, so data sets read from readers must be cleansed.
- Spatial and temporal \[1\]: By tracking and monitoring tagged products, we can obtain information of location and time change. So expressive data model suitable for application level for interaction which dynamically generates changes of location and time is required. However, the massive amount of data resulting from the proliferation of RFID systems pose some interesting challenges for data management recently \[8\].

In our suggested system, we divide EPC into three parts such as the code of item, the code of order number and the number of sequence to make full use of each data and facilitate information retrievals from only the RFID tag data in the system.

2.2 Process Control and Its Range

Several working processes are involved in completing a finished product using raw materials and subsidiary materials. There are constant sequences and orders in working processes, called manufacturing process. As most enterprises have the combinations of very complex manufacturing processes, the partial process errors can not only damage the whole working processes but also affect manufacturing products critically.

The aim of ‘Press Control’ is a smooth management of all production process by controlling or regulating each work process based on production result according to definite time plan \[6\]. The process control mainly consists of following procedures.

- Procedure planning: Determine time and place needed for each process. Determine process sequence and necessary manufacturing process for each product.
- Schedule planning: Arrange details of production such as time required or due date of delivery. Calculate the number of days needed to carry out the plan and allocate the actual date in calendar.
- Process control: observe progress of each working process or development or change of process according to the schedule.

Process control plays an important role in enhancing productivity through segmentation and standardization of each working phase. In this paper, we propose a method for effective process control in the above mentioned process control procedure.

2.3 RFID Cuboid and PathTree to RFID Flow Analysis

2.3.1 RFID Cuboid

Various researches have been conducted in the field of data flow analysis. Jiawei Han et al showed a new method for data flow analysis in terms of data compression called “RFID Cuboid” \[3\]. This is the core idea in solving problems in the current system. We will utilize such RFID’s characteristics as data generalization, merge or separation of path segments. The RFID-Cuboid is composed of three tables:

- Info table: stores product information for each RFID tag,
- Stay table: stores information on items that stay together at a location,
- Map table: stores path information necessary to link multiple stay records.

In our system, we mainly exploit the modified path-tree to construct Stay table-like structures, and use the existing data sets as Info table because the existing information data in the ERP system is already structurally well-formed to represent information of each item so that we can use them directly. But we
do not define Map table additionally because we have to manage the information of each object movement, not chase the grouped items movement.

2.3.2 *PathTree*

In this paper, we utilize a PathTree, a new data structure used in RFID Cuboid in order to efficiently manage real-time production process control. PathTree proposed in [3] is based on bulky object movements. Since items move not individually but in a group of specific unit, management of items by group becomes necessary. To do this they used a GID (Generalized ID), a flow relation between a super ordinate node and a subordinate node in the tree. In this way they proved the efficient query processing and improvement of level of data compression through data generalization.

However there are several problems in using a PathTree directly for the process control in the system, due to difficulty to control individual information of each item, since the minimum management unit is by group.

- Rather inefficient to apply at the small and medium enterprises because items frequently move individually. However, it is helpful in the large conglomerates where items move by group.
- Inadequate for the process control such as tracking of previous process, detecting current process, and estimating next processes in production process of item.
- Unable to analyze the movement of items or error in the production process because it is difficult to track flow of items.

In an effort to overcome above difficulties, we suggest a functionally modified and complemented PathTree, called Procedure Tree.

3. RELATED WORKS

In this section, we will introduce mining technology for RFID data sets in the real world and the process control cases through the ERP systems. We will also examine the problems of the current system and suggest how to overcome the existing problems.

3.1 Mining RFID Data Sets

Before considering the research topics associated with RFID data mining, we note that there has been a great deal of interest in the idea mainly within the company over the past few years. In [3], the authors show five categories in mining RFID data sets.

- RFID data cleaning by data mining [16]
- RFID data flow analysis
- Path-based classification and cluster analysis
- Frequent pattern and sequential pattern analysis [9]
- Outlier analysis in RFID data [15]

Among five categories, we will focus on the second topic, which is RFID data flow analysis. We will introduce detailed drawings and some researches in the following sections.

3.2 Process Control in the RFID-based EIS

Using RFID-based EIS, we can easily manage asset information of a company such as the location and inventory of each product in specific warehouse by attaching tags. Moreover, by integrating both RFID and EIS systems, we can make best use of their respective advantages utilizing these benefits to the production process control as well.

In some countries, cases of applications on this area for the production process control have been founded. We summarize the exemplary case in Table 1 and describe a gap between our aim and potential problems. As you can see in Table 1, although many researches in both the industrial and academic fields have been done in this area, there are few applicable systems that can be used practically. The reason is that securing sufficient amount of data that can be used in the actual environment is hard to come by. The system is also on executing limited experiments in limited environmental scenario such as insufficient number of items that can be located simultaneously in real environments [7].

Table 1. Applications of RFID-based EIS for production control

<table>
<thead>
<tr>
<th>Author</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kiuk K. et al [10]</td>
<td>The real-time shoes product control system using RFID (-Works only in experiment, not considering real working environment in practice)</td>
</tr>
<tr>
<td>Okjae L. [12]</td>
<td>Apply RFID system to semi-conductor manufacturing process package (-Needs user awareness as to which stage is processing now and system cannot estimate the next stage)</td>
</tr>
<tr>
<td>Injung P. et al [13]</td>
<td>Master card-based processing to manage production (-Uses human hand to verify with the card one by one (waste of labor force))</td>
</tr>
<tr>
<td>Jiawei H. et al [3, 4]</td>
<td>Assumes that items usually move together in large group (not fit to manage each item respectively)</td>
</tr>
</tbody>
</table>

3.3 Problems Remaining in the Previous Researches

As we can see in the previous section, the existing researches still have problems to be solved. In other words, few cases are readily applicable to actual environment. Since many experiments have been carried out in the limited environments, certain assumptions or premises must be accompanied. We next define the challenges which need to be solved as following.

- What is the present status of a real-time inventory of each item in each process?
- What is the average time needed from a specific process to an intermediate or a final process? Which process takes up the most time?
- How do we detect the position of the current process and predict the next process within the entire workflow? How many processes are left?

From the manager perspective, the first problem is one of the most important tasks to manage the entire production flow in real time. The sales department can obtain real-time information of the procedures from orders to deliveries with the information. Moreover, as soon as the manager detects any problem during the manufacturing process, he can fix it promptly. The second problem is closely related to the efficiency of production process.
As the manager can check the average time required for each phase, he can try to find a solution to reduce the most time-consuming stage and improve the productivity by modifying the processes. The last problem is to predict the stage of workflow without human judgment. This leads to the automation of process control. For example, when an operator wishes to obtain the information of an item, he can check the current status of an item promptly even though he doesn’t have the exact information of the item. As the system can overlook the entire stage of each item, it also can send each item to the next stage correctly, detecting some problematic situations such as missed or duplicated stages in producing.

In the following section, as a solution of above problems, we propose and implement a novel system with a ‘procedure tree’, which is an improved form of PathTree.

4. REAL-TIME PROCESS CONTROL

As the main part of this paper, we present the architecture of our suggested system in detail. We will also analyze the data sets used in the current system and then propose an efficient methodology to manage inventory in each process using a procedure tree.

4.1 The Architecture of the Suggested System

Compared with the previous system in [2], the striking characteristics of the suggested (Figure 1) system are as follows.

- **ERP databases as the substitute of Info table**
- **Addition of management table for path information**
- **Extended column of tag status’ function in the RFID database**

In the suggested system, two new tables have been added to store and load path information. We extend the role of ITEM_STATUS in the tag information table to easily verify inventory of each process. In the existing system, we used ITEM_STATUS to represent the status of tag itself such as ‘0’ indicating the tag has been published, ‘1’ or ‘2’ indicating the tag is waiting be warehoused or already warehoused and so on. However, in the system, we change this data into the label of edge in the procedure tree so that a manager can check the status of the current item and thus he can predict what the next process will be. The details of each table will be described in the following section.

4.2 Process Control with Procedure Tree

4.2.1 Analysis of the current standard data sets and preprocessing

Before performing the inventory management in each phase, we analyze the standard information data sets in the ERP system. As shown in Table 2, there are 3,051 items produced in the company. Each item requires several complex processing stages to end up as a finished product. In the current database system, in order to maintain i-th stage of the total stock that has n number of processing stages we need n number of duplicated data (1 ≤ i ≤ n). It is likely that we are forced to preserve a large amount of original data that has a little chance of reuse. Rather, it is more important for the manager to obtain the information of real-time manufacturing process flow or future process than the ones that have already processed in the past. This means there are many similar cases that have such redundant data that need to be removed or compressed.

<table>
<thead>
<tr>
<th>Number of procedures</th>
<th>Number of items</th>
<th>Number of tuples</th>
<th>Reduced rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1,094</td>
<td>1,094</td>
<td>100%</td>
</tr>
<tr>
<td>2</td>
<td>903</td>
<td>1,806</td>
<td>50%</td>
</tr>
<tr>
<td>3</td>
<td>806</td>
<td>2,418</td>
<td>33%</td>
</tr>
<tr>
<td>4</td>
<td>202</td>
<td>808</td>
<td>25%</td>
</tr>
<tr>
<td>5</td>
<td>36</td>
<td>180</td>
<td>20%</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>54</td>
<td>17%</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>7</td>
<td>14%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>3,051</strong></td>
<td><strong>6,367</strong></td>
<td><strong>48%</strong></td>
</tr>
</tbody>
</table>

The reduction rate at the right side of the Table 2 shows how data storage space can be shrunk when compared with data processed one by one for each process and data processed as a flow group. Assume that a certain product consists of n number of process. In order to show progress information of each process, we need n duplicated data. However, if we manage based on the flow system, we only need to have a single data required at this point. So, the amount of data needed is shrunk to 1/n.

We also summarize information of each process and workshop as shown in Table 3. From the table, we can verify that processes that are similar or have a continuous working process are done at the same site or nearby work places. The reason is that most companies cannot afford to have respective workplaces for each process. Moreover, this can cause frequent mobility of the products making work very inefficient.

### Table 2. Data analysis of the current ERP database (1)

<table>
<thead>
<tr>
<th>WorkShop</th>
<th>W01</th>
<th>W02</th>
<th>W03</th>
<th>W04</th>
<th>W06</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>4</td>
</tr>
</tbody>
</table>

Figure 1. The architecture of suggested system

- ERP databases as the substitute of Info table
- Addition of management table for path information
- Extended column of tag status’ function in the RFID database
After analyzing the data, a concatenation of the stage name and the workshop is needed to be applied to procedure tree more easily. Otherwise, each node in the procedure tree should have at least one or more information. So, we define a WP_CD.

**Definition 1 (WP_CD, a unique code for procedure in a specific workshop)** Let \( W \) as a set of warehouses and \( P \) as a set of procedures and then let \( P \times W \) denote the set of all ordered pairs \((p, w)\), where \(p \in P \) and \(w \in W\). We call \( P \times W \) as the Cartesian product of \( P \) and \( W \) and each element in \( P \times W \) as WP_CD. We can intuitively see that the maximum number of \( P \times W \) is always less than \(|P| \times |W|\), where \(|P|\) and \(|W|\) are the number of possible procedures and the number of possible warehouse, respectively.

For example, if a “10” procedure is processed in a workshop named “W02”, we name this stages as (10, W02) abbreviated shortly as “10.02” in the suggested system. If we construct \( P \times W \) set using the same method, we can make 24 pairs from Table 3. We use these WP_CDs to construct procedure tree to manage each process.

### 4.2.2 Constructing Procedure Tree

Before constructing a tree, we define a ‘Procedure Tree’ and provide several strong features.

**Definition 2 (Procedure Tree)** A procedure tree, denoted by \( PT = (N, E) \), is a tree-like structure that has information related to process control, where \( N \) is a set of nodes that consist of a WP_CD and the average time consumed in each process and is represented by the size of a node (seconds); \( E \) is a set of relations between two nodes with each label on each edge.

To represent a \( PT \) graphically, a node \( n_i \) is drawn as a circle with a WP_CD and by a specific size which presents an average time from an \( i \)th stage to the next stage. An edge \( e_j \) is drawn as a directed arrow between nodes from \( n_i \) to \( n_j \) \((1 \leq i \leq j)\) with a PID (Procedure ID), which is a kind of GID [3, 4] in the PathTree. The maximum height of \( PT \) is equivalent to the maximum number of phases. Additionally, an average cost time, denoted by \( C_\text{avg} \), computes as follows. In the equation, the cost value \( C_\text{avg} \) is the average cost until just before, \( n \) is the number of cost inputs and \( C_k \) is the difference between the current time and the lastly modified time.

\[
\text{average cost} = \frac{C_\text{avg} \times n + C_k}{n + 1} \quad (n = 0, 1, 2, \cdots)
\]  

A procedure tree has following predominant characteristics compared with PathTree:

- In the procedure tree, we can manage information of each item respectively, not bundle of items, as the tree allocates a relevant PID, which has an equivalent role to GID in PathTree, with respect to the current location of each item.
- Based on the production plan of the item, and managing final phase of PID separately, we can manage such process control as detecting the current process, tracing the previous process and predicting the next one in the procedure tree.
- In the PathTree, the consumed time is represented as *time in* and *time out* while in the procedure tree it is represented by the size of each node. So, we can figure out the most time-consumed process as well as average consumed time in each phase without any computing.

Once we analyze and preprocess the data, as shown in previous section, we can construct a procedure tree which is a modified PathTree structure. At first, we have to group by each item and trace a track of the full path to be a completed product. For example, an item whose code is “107670004” has three stages, as Figure 2 illustrates. In this case, we can write a full path of the item as a form of an ordered set \{10.01, 20.02, 30.02\}. Each element in the set is the label of each node and each average time cost of stage determines the size of each node dynamically. The flow from one node to another is an edge and the label of each edge is sequentially numbered. By applying the same way iteratively, the procedure tree of above data can be created as visualized in Figure 3. We can also add new procedure sets easily with inserting new nodes under the parent node corresponding to relevant stage if a company creates a new production process for a new product.

- Figure 2. A full path of an item (107670004)

- Figure 3. Construction of a procedure tree

After constructing a procedure tree, by using DOM API, we convert this graphical data to XML format (Figure 4) for data exchange between heterogeneous application program and real-time showing on the web. We also store both the full and partial path of each item, PID and time consumed \( (C) \) in their respective database (Figure 1). A client user can obtain this XML file from
the server or path information from database. Then, he or she performs inventory management in each stage with constructing a procedure tree based on the file or the database.

Figure 4. XML format of constructed procedure tree

Algorithm 1 summarizes the simplified descriptions for constructing a procedure tree from the data sets related to stages and how to store it in the database. As the input value, the code of each item and partial path set of the algorithm is a procedure tree and then inserts the partial paths and the full-paths of each item in the form of (ITEM_CD, FULL_PATH) and (WK_PROC_CD, PATH) into the tables, respectively.

Algorithm 1

<table>
<thead>
<tr>
<th>Input:</th>
<th>Each path in each stage, Code of each item</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output:</td>
<td>Procedure tree, Full-paths of each item</td>
</tr>
</tbody>
</table>

Method:

```csharp
1: if (Validate()) { // Validation
2: while (true) {
3: // Partitioning,...
4: if (DataSet = Partitioning()) break;
5: FullPath = ToStringPath(DataSet);
6: // Partitioning,...
7: // Decide whether if the path is (non-)sequential
8: ITEM_CD = new StringTokenizer(DataSet[0],
9: "\"\").NextToken();
10: IsSequentialPath = ProcedureValidation(DataSet);
11: if (!FindProcedure(DataSet)) {
12: AddNode(DataSet); // add new procedures,...
13: AddNode(DataSet); // add new procedures,...
14: // Count sequential paths,...
15: if (!IsSequentialPath) CounterSequentialPath++;
16: Counter++; // Count added path,... }
17: }
18: } // Validation
```

4.2.3 Production process control in each phase using the constructed procedure tree

When using procedure tree, we can easily obtain real-time inventory information per each process because we only have to check the number of items that has identical PID. Moreover, we can figure out information of the previous process of the current process and future process. We can also obtain such information as how long it would take for an item at this point to become a final product and how many stages are left to be an end product.

In the suggested system, in case the user tries to find previous process of the current process or the next process, the system can respond to the user constructing procedure tree based only on the full-path of the selected item and only minimum information of nodes. For example, query could be: “If the PID of an item is 12.1.1., what is the next stage for the item?” (See Figure 3) The answer for this query can be found immediately by searching child node corresponding to the current PID at the procedure tree. In this case, as the item has a full path PID as 12.1.1.0, the system will find 12.1 as a previous stage which is a parent node and 12.1.1.0 as a next stage which is a child node. In other words, the system has an ability to perform context-aware production management in which it recognizes how previous process went, how the next one will progress and if the current process is going accurately.

5. PERFORMANCE STUDY

Before evaluating the performance of our study and implementation of the suggested system, several matters should be considered. In this paper, we implemented this system with the paper manufacturing company, Korea Computer Form (KCF), which mainly produces goods such as a bill of expenditures, specifications, or an itemized account and so on.

As data sources, we used data sets in the real ERP database which consists of a procedure code, a minimized sequential information set, a warehouse code and an item code. By using this data, we create a full procedure tree and store in the RFID database server to get the constructed tree easily and rapidly. In case there could be a situation where a new process of an item being created, we also offer some tools and functions to be able to insert them into the tree easily.

5.1 System Implementation

In this paper, we use C# development toolkit and RFID technology for the implementation of the proposed system. The software environment is Microsoft .Net Framework 2.0 with SQL 2000 database server. And on the RFID side, an ER9501 RFID development toolkit and PM4i RFID handheld reader (Intermec Co.) are selected. The selection is based on support from diverse libraries and reputation of better performance compared with other equipments.

The core function provided by our system is “real-time process control”. We will give a full detail of the system in the next section.

5.1.1 Real-time process control

In order to understand how process control is performed within industries, we need to check some points to be considered. On the server’s perspective, the server collects procedure data sets from ERP database and automatically constructs the initial procedure tree based on Algorithm 1. That is because the function of the mobile RFID handheld reader is less compared with the server, so the server takes over this time-consuming process and burdensome process. After constructing initial procedure tree from the server, the RFID handheld reader constructs a sub procedure tree bringing PID set minimized based on the full path and information of the selected item.

By using a sub tree constructed at the server, the mobile RFID reader can find not only the current process but also the previous and the next processes by searching nodes of the children and parent of current PID. If the current stage is processed and has only one process to be preceded, the reader computes the average cost time between the previous and next stages by using equation (1) and saves in the database (Figure 5). If the current stage has more than one available process (If the current process has more than one child in the tree), the mobile RFID reader shows the user the screen as to how the process would be progressed and asks to select the screen. All processes in the production system are managed in the same fashion. We guarantee consistency of data.
between ERP and the suggested system by connecting ERP implemented in the previous system.

![Figure 5. Process control on RFID handheld reader](image)

Additionally, a manager can easily obtain the amount of each process by only counting the same PID corresponding to the item code.

### 5.2 System Evaluation

We validated the suggested system in terms of time performance. The background computer is a PC with Intel Pentium CPU 2.4GHz, 2GB RAM, running MS Windows XP. The server executes two most time-consuming tasks: initialization of databases and construction of the whole procedure tree.

To evaluate the performance of the suggested system, in the next paragraph, we define databases to use and verify the following criteria in the real world.

#### 5.2.1 Datasets

We collect a huge amount of data in the ERP database as experimental datasets and analyze the data as illustrated in Table 4.

<table>
<thead>
<tr>
<th>Table 4. Analysis of datasets</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Period</strong></td>
</tr>
<tr>
<td><strong># of data</strong></td>
</tr>
<tr>
<td><strong># of paths</strong></td>
</tr>
<tr>
<td><strong># of PID</strong></td>
</tr>
</tbody>
</table>

As you can see intuitively, there are so many items that are sequentially processed in production management system (98.77%). On the other hand, a few items have non-sequential path (1.23%) from a total amount point of view. Paths (PIDs) are divided into two categories: sequential (49.74%) and non-sequential (50.26%) paths. Although there are a few number of non-sequential path item, however, we can see peculiar point that the weight of sequential paths is on the parity with that of non-sequential paths. It could be occurred when an enterprise modify the standard information of production process of item or its order for productivity improvements or efficiency of machines in the company. We apply these data sets in experiments to evaluate our proposed system.

#### 5.2.2 System Evaluation

To verify the suggested system, we condense the results of the empirical experiments into Table 5. The server load in the suggested system is reduced drastically, due to modification of data sets for only current phase, decreasing transactions such as readings and writings in databases to do process control, while the previous one processes whole data sets corresponding to the number of processes. Furthermore, in the data itself point of view, the suggested system can track the trace of the previous process, find the current process, and infer the next processes because the system has dependant and continuous data sets, showing relations between data sets in the procedure tree, while the previous one has independent and discrete data sets without interoperability.

Additionally it means that the system can easily detect the process errors such as duplicated processes or missing processes in the process control so that we can make alternatives instantly. In the item management’s point of view, as the previous method deal with information of each process by group, it cannot manage information of each item individually. On the contrary, the proposed method can manage items one by one as well as grouped items.

So, with the suggested method, we can perform real-time process control and inventory management efficiently and effectively, which cannot be done in the previous one (illustrated in Table 6).

<table>
<thead>
<tr>
<th>Table 5. Running time of each device</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Phase</strong></td>
</tr>
<tr>
<td>Initialization (only 1 time)</td>
</tr>
<tr>
<td>-Gathering data sets from DB</td>
</tr>
<tr>
<td>-Constructing PT</td>
</tr>
<tr>
<td>Search</td>
</tr>
<tr>
<td>-Sub PT construction from DB</td>
</tr>
<tr>
<td>-Find previous phase</td>
</tr>
<tr>
<td>-Find current phase</td>
</tr>
<tr>
<td>-Find next phases</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 6. Comparison between the existing system and the suggested system</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Previous System</strong></td>
</tr>
<tr>
<td><strong>Database</strong></td>
</tr>
<tr>
<td><strong>Server Load</strong></td>
</tr>
<tr>
<td><strong>Process Control</strong></td>
</tr>
<tr>
<td><strong>Impossible to detect production errors</strong></td>
</tr>
</tbody>
</table>

### 6. CONCLUSIONS

RFID, a substitute for the barcode, is a wireless identification technology widely used in the various areas such as distribution industries and production control management. Many enterprises are making efforts to apply RFID technology to real business environment in order to adapt to the rapidly changing environment where rapid renewal of information becomes
necessary. A development case has been reported in which the existing RFID system has been integrated with the EIS. However, these systems have some limitations in showing mobility and production process of the items dynamically because the system only uses such RFID’s characteristics as wirelessly reading multiple items simultaneously. It is also inefficient to manage such time-related dynamic information as material requirement planning (MRP) and production forecasts.

To solve the above problems we proposed a ‘procedure tree,’ an enhanced and modified version of PathTree, developed for efficiently performing real-time process control in the enterprises. By using the ‘procedure tree’ we suggested a new method of real-time process control.

After implementing the suggested system connected with RFID-based EIS, we were able to estimate or track process flow which the existing RFID-based production management systems had difficulty executing the task. Because the system is by itself capable of predicting how the current process will progress or next progress without the aid of human decision or judgment, it can detect such process errors as overlapping or missing processes easily so the operator can take appropriate actions immediately. In sum, by using the suggested method, dynamic management in all areas in which prediction is necessary including sequential flow or tracking and managing of all time-related process as well as normal process control becomes possible.

So far the application of the suggested system in this paper has only been confined to the area of process control. However, it is imperative that the suggested method in this paper should be developed as general-purpose system so that it can be applied to all enterprises which have sequential flow system so as to manage their process control dynamically.

7. REFERENCES