การพัฒนาระบบสารสนเทศเรียลไทม์ในอุตสาหกรรม

Development of Real-Time Information System in Manufacturing

By

Mr. Kosol Chakanjsilp

ID 39067019

Advisor

Dr. Dentcho N. Batanov

A PROJECT SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE SYSTEM DEVELOPMENT PROJECT

MASTER OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY

SECOND SEMESTER ACADEMIC YEAR 1998

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ชื่อหัวข้อ	การพัฒนาระบบสารสนเทศเรียลไทม์ในอุตสาหกรรม
นักศึกษา	นายโกศล ฉกรรจ์ศิลป์
อาจารย์ที่ปรึกษา	Dr. Dentcho N. Batanov
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
พ.ศ.	2540

บทคัดย่อ

การพัฒนาระบบสารสนเทศแบบเรียลไทม์ในอุตสาหกรรม มีวัตถุประสงค์เพื่อศึกษาและพัฒนาระบบสารสนเทศในการเฝ้าดู (Monitoring) ประสิทธิภาพ, คุณภาพ และผลการปฏิบัติงานในระบบงานสายการผลิตในอุตสาหกรรม โดยใช้หลักการทาง Client/Server ในการออกแบบ ซึ่งจะสะดวกต่อการพัฒนา (Development) และง่ายต่อผู้ใช้เพราะการติดต่อกับผู้ใช้เป็นกราฟิก

ผลการศึกษาพบว่า การใช้ระบบสารสนเทศแบบเรียลไทม์ในการพัฒนาโดยใช้หลักการ Client/Server และการจัดตารางเวลางาน (Time Scheduling) สามารถแก้ปัญหาในระบบงานที่ต้องการความรวดเร็วในการรับรู้ข้อมูลข่าวสารเนื่องจากงานอุตสาหกรรมเป็นงานที่มีการผลิตตลอดเวลา การมีระบบข่าวสารแบบเรียลไทม์ทำให้สามารถตรวจสอบจุดผิดพลาดในสายการผลิตและทำการแก้ไขได้ทันกาล ก่อนที่ปริมาณของเสียจะเกิดมากขึ้นจากการไม่มีระบบข่าวสารที่รวดเร็ว การประยุกต์ใช้ระบบสารสนเทศแบบเรียลไทม์ในอุตสาหกรรมการผลิต จึงให้ประโยชน์มากมายและทำให้การผลิตมีประสิทธิภาพมากขึ้นอย่างมหาศาล

Title	Development of Real-Time Information System in Manufacturing
Student	Mr. Kosol Chakanjsilp
Advisor	Dr. Dentcho N. Batanov
Level of Study	Master of Science in Information Technology
Major	Information Science
Year	1997

ABSTRACT

Purpose of the development of real-time information system in manufacturing is study and develop information system for monitoring efficiency, quality and performance in shopfloor process of manufacturing. Using the real-time scheme and technique of client/server architecture which gain in development time, the use of informations in organization and user-friendly with graphical interface feature.

The result can be used to correct problem in continuous process which need fast information to detect cause of problem and performance of shopfloor process. System led benefit of fast corrective action on shopfloor's problem which defect will grow fast and lost more if no have real-time information system.

CONTENTS

CONTENTS (cont.)

LIST OF TABLES

Page

Table

LIST OF FIGURES

Page

Figure

LIST OF FIGURES (cont.)

Page

CHAPTER 1

Introduction

1.1 Background

The use of computers in manufacturing brought many productivity gains, but modern manufacturing system have become extremely complex to manage. The menufacturing scene is beset by constant changes in technology, product and production volume mix. Manufacturing industries have tried to automate production and its management at varying levels of control and financial investment with miexed success.

A popular view is the full potential of productivity improvements made possible by modern electronic computing technique is not being realized because of the lack of a systems approach capable of integrating the islands of automation which tend to proliferate with progress in manufacturing engineering while allowing each information to function to its fullest extent. This has led to the idea of designing systems with integration through computers as the main theme.

Client/Server technology has the potential to provide the conceptual and computing philosophy needed to achieve integration of the manufacturing enterprise. In addition to helping solve specific problems, it also has the potential to change methods of computing in other domains in the future.

A common and significant impediment to the timely flow of infomration within and between enterprises, and therefore an impediment to more agile manufacturing businesses and industries, is the costly, massive effort required to implement and integrate information systems that share realtime manufacturing data throughout organizations.

On the factory floor, realtime data collection from shop floor devices and personnel, realtime data processing and validation, realtime data sharing, and realtime application interoperability suffer, even in the most automated plants. Most realtime information is not cost effectively made available to the people and applications that need it.

Factory-floor systems which focus on the operation of produciton equipment and on the control of processes, communicate neither directly nor regularly with front-office information systems dedicated to accounting, forecasting, and other business reporting and planning activities; or with design and engineering systems. Upstream information systems are unaware of important manufacturing details, such as the availability of appropriate tools, labor and materials, maintenance schedules, records of past process performance or the status of work in process.

Middle-level information system know as Manufacturing Execution System (MES), bridge the critical informamtion gap between upstream, transaction-oriented, planning activities and downstream, event-oriented execution activities. MES application track and manage all aspets of a job on the shop floor, at any point in the production cycle, in near realtime. MES applications play essential supervisory and monitoring roles that link all levels of manufacturing operations. For example, they identify bottlenecks and material shortages on the shop floor, and they provide up-to-minute process performance results along with comparisons to past performance and to projected business results. Manufacturing efficiency, the goal of most Enterprise Resource Planning (ERP) systems today, is not enough. Manufacturers need more. Speed requires new methods and new applications that can optimize the entire order delivery and customer service process, not just the manufacturing component.

Realtime information system is one part of MES system which use the nearly realtime schema to process shop floor data into execution information. Information was distributed across organization via networked enterprise. The ability to access information, integrate solutions and communicate information across networked enterprise and customer applications in fast speed is vital. This research and development will introduce the project which was designed and developed on realtime information system in harddisk industrial. Shop floor information was introduce to offfice and management level in nearly realtime speed.

1.2 Objective

1) To design and develop in-house realtime information system in nearly realtime manner.
2) To develop application in realtime mode for monitoring shop floor process using Client/Server approach.

3) To study potential of technology which suit for manufacturing in the future.

1.3 Stage of study

No.	Desciption	Oct'97	Nov'97	Dec'97	Jan'98	Feb'98
1	Analyze production process	■				
2	Investigate user requirement	■	■			
3	System analysis and design		■			
4	Design data flow and database		■			
5	Develop application		■	■	■	
6	Test and debug				■	■
7	Implement in manufactoring process					■
8	Conclude					■
9	Documentation					■

Table 1.1 Stage of study

1.4 Expected benefit

1) We can use the designed to apply with other levels of production processes.

2) Application can track the performance of production in realtime mode.

3) Managements and engineers can use this tool to monitor and capable on fast problem solving.

4.) Equipments and devices in shop floor process can be monitored its performance remotely.

CHAPTER 2

Client/Server Concepts

2.1 Introduction

Client/server concepts is a logical relationship between an entity that request a service that is client, from another entity (a server) that provides a shared service as requested. The client and server parts may or may not exist on distinct physical machines. The relationship between a client and server is conducts by means of transactions consisting of well-defined requests and responses. Client/server relationship are often referred to as cooperative processing.

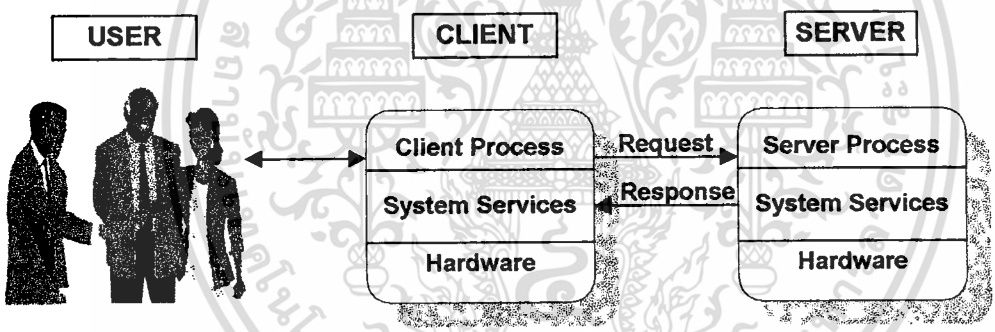


Figure 2.1 Client/Server system.

The term client-server has changed over time, from the old hardware-based of a host connected to multiple terminals, to th emore recent software/application model which utilizes multiple machine to distribute processing. This more current model focuses on spliting the workload of a process across multiple machine to increase overall efficiency. This is accomplished b networking multiple processing clients to one or more data servers. The object is to leverage the processing power of all machines on the network without forcing one to handle too significant a load. Like many technologies, client/server can be implemented in several different ways.

Client/Server model in thie project consists of hosts (file servers and database server), client workstations and batch/background processors. The database servers are specialized compuers designed to safely store and maintain business data in a secure but readily available environment. They utilize open architecture and network protocols to provide secure access to stored data.

This project uses activity servers to perform batch and background processing. These machine, in essence, are extremely fat client workstations, in that they execute specific program against a database to manipulate its data. The activity server waits until it it told to execute a program and then does so in the background with no user intervention. The primary advantage of this architecture is the ability to off-load activity and report processing to an unattended machine on the network. This machine is very self-sufficient and needs very little human interaction to function. It is basically set up once and left to perform its role. Overall, this has a great positive impact on system effectiveness and efficiency.

2.2 Client/Server benefits

1. **Cost management and reduction**

A client/server environment lends itself to using graphical user interface (GUI) applications, which are easier to both learn and use than host-based computing applications. This reduces the training cost involed with the application. Because the application is more intuitive in nature, users quickly become proficient and increasingly productive. In addition, costs are reduced with the ability to use multiple platforms; smaller, faster and cheaper servers and retrieve and manipulate data more efficiently.

Although techgnology costs may increase with the implementation of client/server, the productivity gains and ease-of-use of the system can make migrating to client/server quickly profitable for the company.

2. **Greater scalability**

A client/server environment is considered scalable-it is usable in different sized cnvironments across different platforms. It can be used in a small office to connect only a few computers on a LAN (Local Area Nertwork) or across the country to connect

LANs located in diferent cities using different platforms, on a WAN (Wide Area Network). Client/Server is flexible enough that adding additional users, cliens, servers, or LANs to the network can be easily accomplished. Computer networks are a combination of hardware, software and wiring schemes that allow dispersed computer system to interact and allow users to share system resources and data.

3. **Better performance management**

Network performance has always been a concern for everyone from the network adminstrator to the end users, to the president waiting on a report for the board meeting. Network performance generally refers to the speed at which computers communicate with each other. The most common way to jude system performance is by the amount of time it takes to perform tasks such as printing, receivring files from a servers, or performing a query. There are many sophisticated ways built into network software to monitor system performance.

In the past, increasing system performance meant adding large expensive host computers, or at least expensive proprietary parts for a pre-existing host computer. A client/server environment affords many options for increasing system performance, including increasing the amount of memory on the server or workstation, adding additional servers or workstations, adjusting configurations and hardware components, or increasing network bandwidth.

4. **Economical resource sharing**

One of the biggest advantages of a networked environment is the ability to share resources, printers, modems, files, data and applications. While it is expensive for a company to equip everyone in the organization with their own printer or modem, client/server environments make it economical for people to access and share hardware devices.

In addition, a database server allows multiple users and processes to access and manipulate the business data at the same time using a variety of tools. Sharing a common database sotred on a server is the most common user for a client/server environment.

Because multiple users in a company need to have access to current data about customers, products, inventory, production stats etc. Storing one central database on a server is more efficient.

CHAPTER 3

Information System Relevance to Manufacturing

One of most popular system in manufacturing is the computer-integrated manufacturing (CIM) which is an effort to integrate activities in manufacturing through the medium of computers. It is now widely accepted that CIM leads to widespread improvement in productivity. Long-term benefits of CIM are based on the realization that CIM is neither hardware nor software for automation, but a strategy. It is a strategy related to computer-based sharing of resources and the availability of timely and appropriate information to optimize performance of the entire manufacturing organization. The major limitations of specifically designed CIM system as follows:

1. The need to reinvent product for the wide variety of manufacturing systems and industry requirements that are encountered.
2. The inability to reconfigure the system when previously unforeseen product and/or process changes are required.
3. The difficulties of achieving inter-company integration where information and functionality are shared across company boundaries.

3.1 Software development aspects of CIM

3.1.1 Complexity and variety

Manufacturing systems are complex and varied in nature. It is impractical to imagine that a single solution or software package will address all the needs of all manufacturing firms. A practical approach to desiging software for managing CIM systems is to build generic solutions to the greatest possible extent, and then to customize them to suit the needs of each firm. Thus, client/server approach development provide a good starting point in the design of practical software.

3.1.2 Abstraction issue

From manufacturing world to computer world

Manufacturing-related people think of their systems in terms of parts, conveyors, testers etc. They think in terms of 'objects'. An object orientation approach allows designers and programmers to construct software counterparts of manufacturing entiies easily with little conceptual mismatch. This makes it easier for system designers to discuss models with users.

Aggregation problem

Not all CIM applications require the same level of details in the description of the system state. Representation of important aggregations for different decisions being made in a factory is a critical feature of good software design in manufacturing. The ability to treat objects as a collection of other objects in visual programming are useful in representing aggregations.

3.1.3 Data management

Database is very important role in storing and retreiving data needed in manufacturing. Currently, relational database are being used extensively in engineering applications. And this project also use relational database to store data.

3.1.4 Hierarchical control

Factory control systems have distinct decision making levels. The number and definitions of levels varies from situation to situation. A popular model consistes of the following levels:

- corporate
- factory
- area (shop or department)
- cell (a collection of workstations)
- workstation (a collection of machine or devices)
- device

In a typical hierarchical model of control, control commands and decisions flows downward, and production status and performance information flow upward. The factory representation is therefore required to operate at different levels depending on the control or decision logic being implemented. As a result, different computer programs or plans are needed to reflect the respective needs, often at the cost of information to other levels. An object orientation approach allows data and procedures to be respresented at different levels of abstraction as appropriate to the situation. Software possible to present multiple views of the situation in a consistent and integral manner.

3.2 Communication issues

Electronic data communication is an important part of modern manufacturing system. Communications among various computers and control devices remain problem areas in the implementation of system. The development of the standards MAP (Manufacturing Automation Protocol) and OSI (Open System Interconnection) contributes to solutions in this area.

3.3 Integration problem

3.3.1 The common database approach

CIM applications for different purpose have been developed independently. As these application have at least partially common data needs, this practice has led to data redundancy and consequently data inconsistency. A seperation of the application logic and the data paved the way for databases where the data needed for different applications is stored in one place. The advantages of this architecture is the reduciton of redundant data, avoidance of inconsistency, data sharing, standard enforcement and integrity maintenance (Date, 1990).

The database provides the common platform or model through which the different applications have their data needs fulfilled. The design that model in a way to accommodate the different views required for all individual applications. If all data are treated in the same way, this model could turn out to be a gigantic system. This could seriously affect the response time if all queries are to be handled by one database.

Work-In-Process (WIP) extracts may be obtained only in batch mode a few times a day. They require prohibitive time for real-time access. Not all applications have the same degree of urgency. While a process engineer may accept a five-minute wait for the result of some yield analysis, the operator requesting advice from the scheduling system expects an immediate answer. The boundary between urgent and not-so-urgent data corresponds to the boundary between current and historical data. Keeping the current image in RAM (Random Access Memory) while having historical data on disk would be one way of handling the problem.

Though this apolicable to most situations, this approach may not solve all problems as some specific applications such as Statistic Process Control (SPC) have unique requirements. SPC requires measurement data from machine to detect undesirables trend in equipment performance.

The convention database approach suffers from other limitations. Database speed does not meet the need for flexibility required by future applications development. All current applications may need to see the workstation capacity as an average value. When developing the database, it is therefore natural to have a workstation table with an 'average capacity' field. The applications will know of this structure and will direct their query about capacity data toward the workstation table. When a new application requireing a finer level of detail, the workstation table must be expanded to accommodate the needs of this new application.

Now, database design keep capacity data both in an aggregate and in a detailed form or use mechanism as view in SQL (Date 1990). A different name must be used for the view and the base table, which may result in modifications of the existing applications. The two data structures may coexist if the detailed information is required for some, but not all equipments. Then the old application programs would have to know whether to direct their query toward a base table.

3.3.2 A common object model

The use of database for integration purposes was the first step in the right direction. A common data moving around the factory, connecting different functions or departments. Since data is raw and needs to be interpreted in different ways. This approach results in much traffic in data and problems.

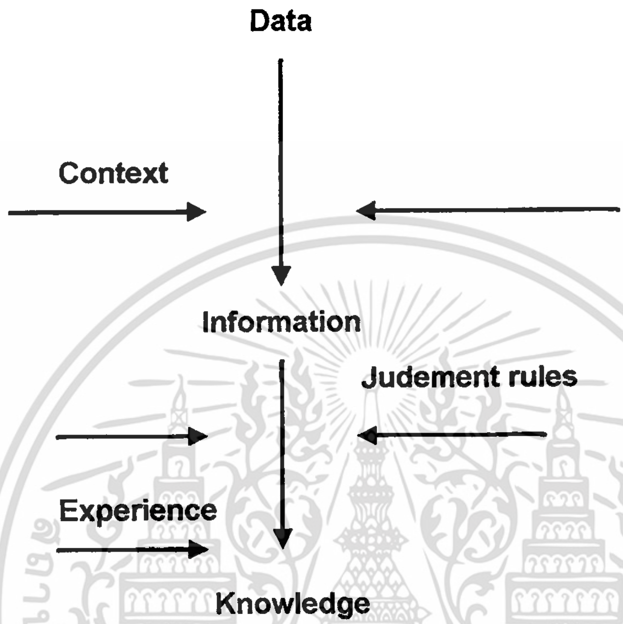


Figure 3.1 Data, information and knowledge

As shown in Figure 3.1, data, information and knowledge can be seen as using different levels of abstractions of the real world. Since an object encapsulates the data and procedures that provide the context in which they are used, an object model is basically an information model of the system.

CHAPTER 4

Realtime architecture for manufacturing application

A computer architecture is composed of two elements, a structure and an organization. Structure describes the components and their interconnections. Organization presents the dynamical appearance of components and their management according to the selected operational principles.

4.1 Conceptual framwork

The specific problems involved in applications such as process planning, production scheduling or controlling process variations require specialized solutions. The task of designing a generic framework is made more difficult because commonly accepted guidelines or speifications for factory automation modules do not exist. Defining a framework is similar to designing a system model.

Framework is based on a specific modularization, or partitioning to serve as a skeleton for building most CIM systems. The modules have been defined at a high level to promote reusability and interchangeability. They reflect the basic system functions that have in most computer system applications in manufacturing.

The primary concern of a manufacturing firm is to transform inputs in the form of raw materials into finished products to be sold to customers. The development of information systems, heuristics and algorithms to make decisions in planning , monitoring and controlling activities as part of this transformation process in manufacturing plants. Most manufacturing systems can be viewed as consisting of two parts, decision and plant or physical systems as Figure 4.1

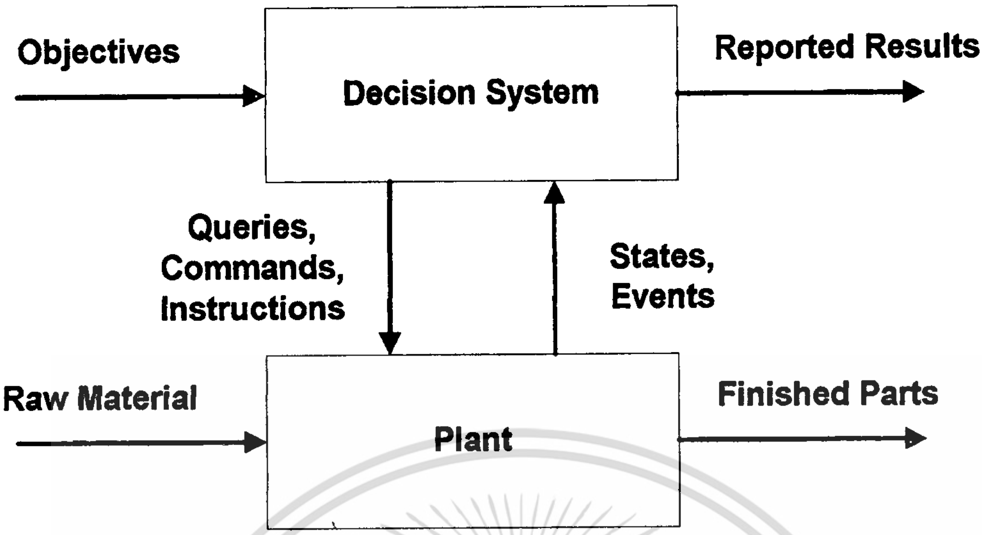


Figure 4.1 A high level view of a manufacturing system

Decision systems include not only automated systems but also manual systems. Plant may consist of human operators or machnines with local or embedded controllers used in the transformation process. The communication from the decision system to the plant is in the form of queries or commands/instructions. Commands are meant for automated devices whereas instructions are issued for humans. The plant or physical system communicates its state or events that take place to the decision system.

The view of manufacturing system as Figure 4.1 can be decompose into specialized components as applicable to computerized application systems. The most of the computerized applications in manufacturing have specific decision requirements. The decision system is a combination of the following modules or subsystems.

1. A computer model of the manufacturing plant. Most application are developed with a particular view of the plant in mind. This model reflects the states of the manufacturing plant or the physical world supplying the context for the decision to be made.
2. Decision or control module. This module contains the decison or control logic used in solving problems in the application. The control logic is expected to work on the state model to achieve the desired action or results. The idea of separating plant from

control has been advocated in work on control theory and practised by mechanical engineers for years. This module supports the decision process.

3. An interface between the state model and the plant or physical world. This module allows one to define the state model and acquire data from the physical world.
4. An interface between the decison module and the plant or the physical world. This module has the responsibility of receiving the requests from users or the real world in general and returning the system recommendations or decisions.

Figure 4.2 shows the interrelationships between modules that make up a typical decision system and the physical system.

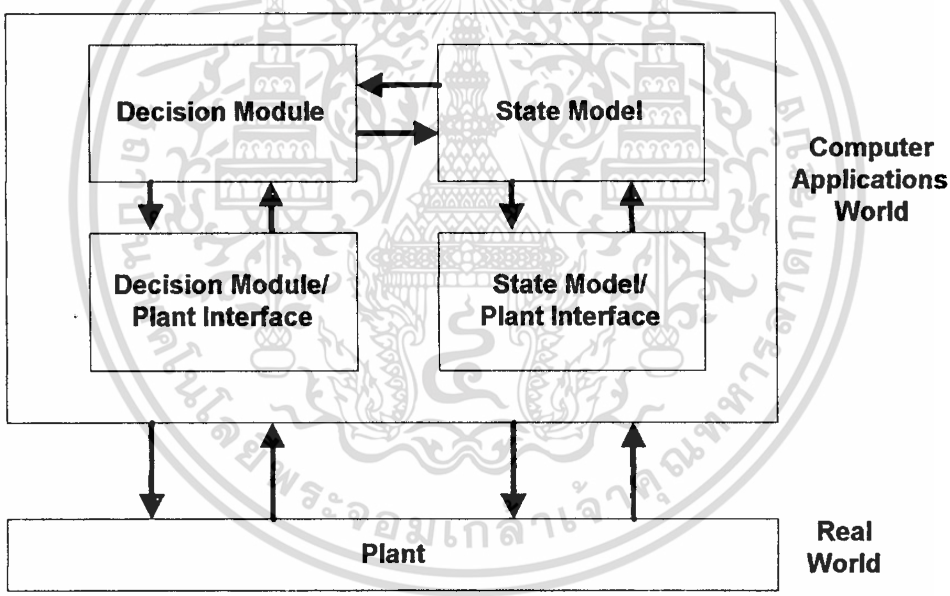


Figure 4.2 Interrelationships between modules

4.2 Decription of the modules

4.2.1 State model module

Object orientation model is the basis for deriving the common state model to be shared between applications. This model instantiated with specific state information and will supply the necessary data for running some applications.

An instantiated object model allows active interaction between that model and the application programs. The application programs need to be informed of the changes triggered by the decisions of other programs or by any relevant changes occurring in the plant. If a measurement result has been entered in the database through the real world/state model interface, the statistical process control (SPC) application should be informed to carry on analysis of the equipment behavior and issue warnings.

And object receiving a message have to send additional messages to inform the relevant application programs. This is to be contrasted with the more static role played by objects that are required only to update or report their states upon receiving a message.

4.2.2 Decision module

The use of the object orientation approach in the modules implementing decision logic is obvious as it was for the state model. An object orientation approach is a natural approach when developing decision modules of a statistical process control modules. The decision module rely only on information at the current operation.

4.2.3 Interface module

The interface is an important part of a computer application as it may influence the success of the implementation. The interface between a state model and the physical world may be manual, automated or a combination of both. An automated interface is found in an assembly line where circuit boards are automatically inspected for faulty insertion of resistors. A mixed interface is an operator using a bar-code reading system to enter the fact that they has completed processing a part. The quality of control logic a system that requires operators to spend half their time inputting data is the success key on the factory floor.

Another desirable characteristic of a CIM system is that different applications should share the same user interface. This has the potential effect of easing the burden on users. Since these applications have different functionalities., the interface to different applications cannot be identical in every detail. The interactive sessions with the user should follow the same pattern. This can be achieved by a system of menus calling the application-specific parts of the interface which should follow some standard for screen presentation.

4.3 Communication between modules

The interaction between modules must be well defined so as to ensure the plug-compatibility of the control modules. Plug-compatibility of modules implies that one should be able to add or remove modules without having to modify the shared model or vice versa. Figure 4.3 illustrate the flow of information taking place between the different modules. The interfaces have been omitted to keep the diagram simple.

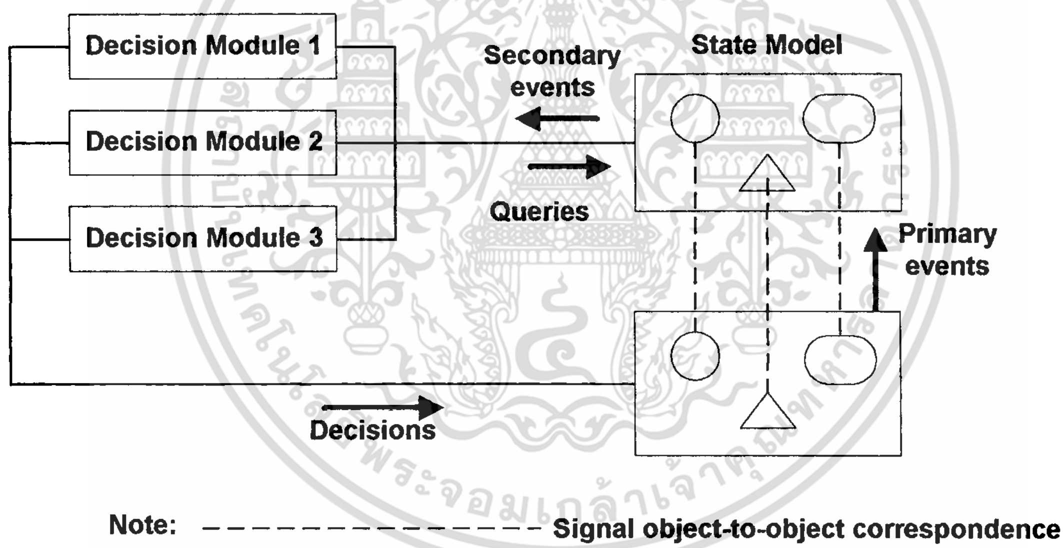


Figure 4.3 Intermodule communication

CHAPTER 5

Design and Implementation of RealTime Monitoring Application

5.1 Introduction

The realtime monitoring applications which was developed in this project is the monitoring system which using in harddisk manufacturing. Client/Server techniques were used to design and implement its monitoring system. The system using a variety data of testers and data entry panals (DNPs) which are all the equipments in manufacturing and need to be tracked status and performance. The testers and DNPs are linked to network system using LAN (Local Area Network).

In applications development, the distributed information was key, not only in fulfilling the technical requirements, but in giving the product engineers and process engineers a formalism that easily captured the expression of their needs and constraints. Real-time manufacturing monitor systems gains quickly action on problem in production process. And useful in using information of shop floor in real-time.

5.2 Background

Harddisk manufacturing production was produced in cleanroom which clean all areas and quipments to protect contamination and control particle with minimal quantity. All testers and equipments which use for test, inspect and measure unit of works were located in this cleanroom. Operators and technical staffs works together in production line which contains many equipments and testers for complete products.

At any workstations, parts will be tested and monitor both by machine and visual inspect by operators. The result of inspection or testing will be collected to summarize as report, in term of process output. The collection of data also use to monitor performance of processes and equipments. With high capacity volume of production cell. Data collection need more automatic or electronic collection for further reporting or analysis. Many equipements were developed to collect these data and linked all equipments together with LAN (Local Area Network).

Information of production will be used by many sections such as production, management and engineering. These sections will shares all informations from same system. Realtime information need to be transparent for all users in all sections. Management use information to manage and plan the next operation of product and for control the performance of product with specific criteria. Production use information for track their performance and througput of the production process. They use information at front-end (in production process) and this realtime information has to response their activities in realtime.

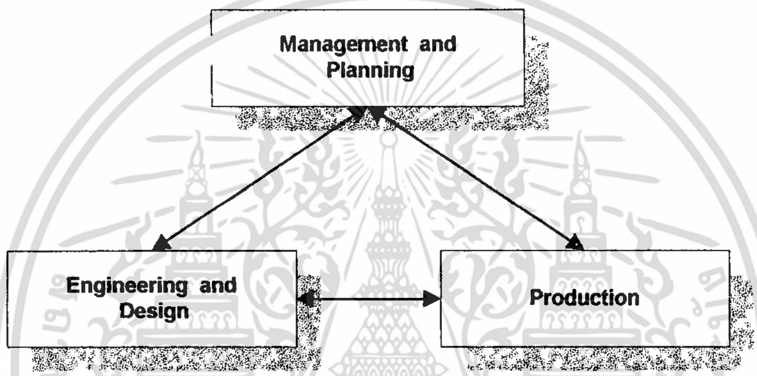


Figure 5.1 relationship of sections who use realtime information.

5.3 System configuration and terminology

There are many entities which were used in the description and analysis in this realtime information system project.

The environment of production process comprise of:

Name	*Meaning*
Product	- model of goods.
Production cell	- cells to build and test parts of model.
Electrical testers	- tester equipment which use for test electrical parameter of read/write head.
Operation Name	- name of operation in each cell to operate specific procedure as work instruction.
Input	- Total input of parts at interest operation.
Output	- Good parts after inspect defect at interest operation.

Defect Code	- defect name which assign to be inspected at each operation.
Yield	- output/input of parts at each interest location or operation.
DNP	- Data eNtry Panel, equipments use for collecting data from production process.
DTS	- Data Terminal System, host PCs use for pooling data from connected DNPs.

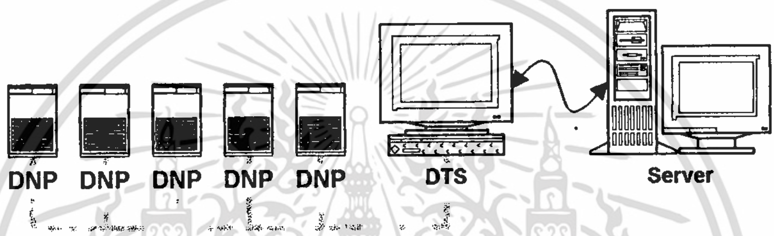


Figure 5.2 connection of DNPs with DTS and network system

Parts will be tested and inspected through mechanical process which DNP unit is the data acquisition system and electrical measurement process which electrical testers are the data acquisition system.

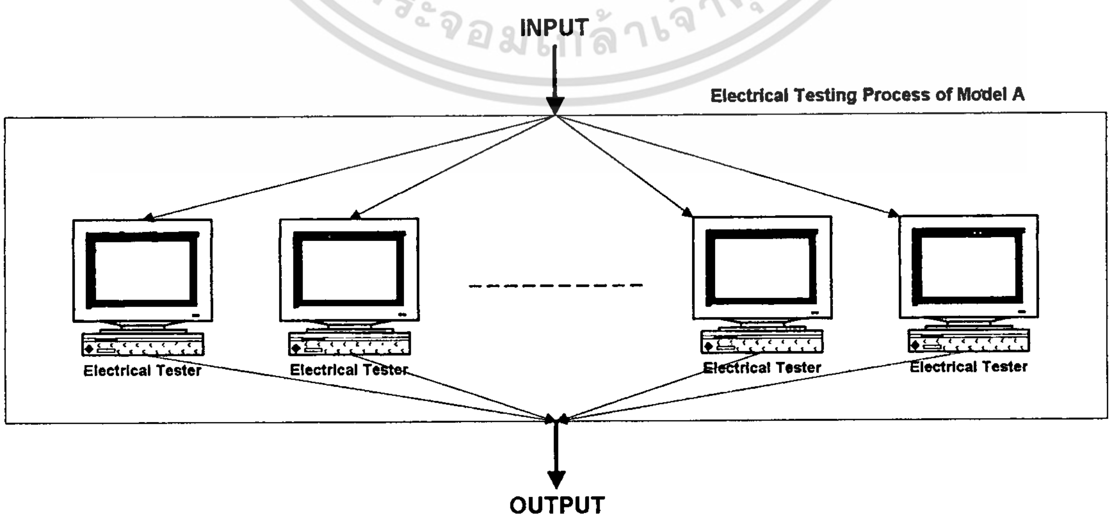


Figure 5.3 Electrical testing process

5.4 Realtime Scheme

The realtime scheme which were considered in this project are the timely consideration which use for trig background processing to process the information.

Maximum validity duration

This mechanism use the fix value of time for trigger. The back-end process will recalculate data with assigned algorithms every identified time interval (z). This time value was designed and defined by consideration of input data flow, network traffic load and realtime interval user need. These all factors was optimized on design process.

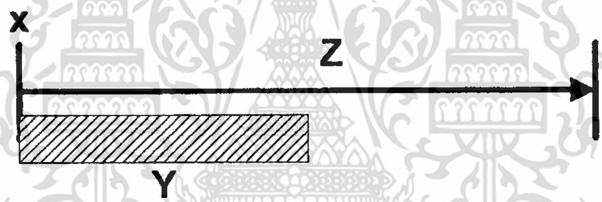


Figure 5.4 present fix time-value mechanism

Figure 5.4 show the relation of maximum validity duration (Z) and processing interval (Y). At the starting time (X), back-end processes will be invoked by time comparison engine, start process the assigned routine with Y period and after end processing, backend-process will set up to sleep state. It will wait until the assigned time value to awake (X+Z). This mechanism will work precisely if processing time (Y) not over the realtime interval (Z). And if execute time of back-end process (Y) is exceed the maximum duration (Z), we can extend this duration to appropriate value if need.

This mechanism let system capable on forecasting load on network, time of validation data and number of back-end process PCs for system.

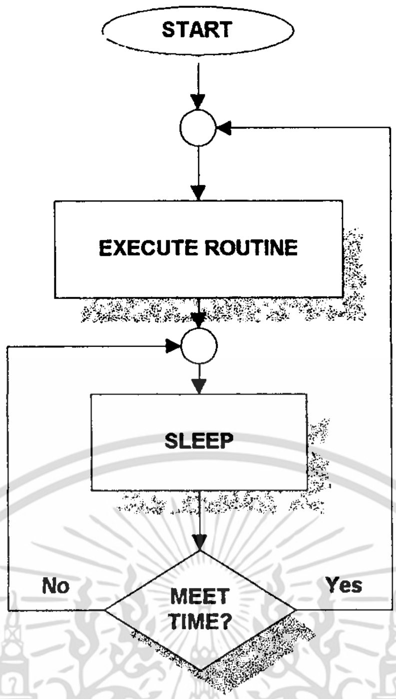


Figure 5.5 diagram of sleep and invoked mode.

Variable of time-value type

In this type of time-value, the amount of realtime range will be adopt follow the current status of network performance and other factor of the system such as the amount of input data, processing loop. The realtime interval (Z) will be adjusted to meet optimum value for system.

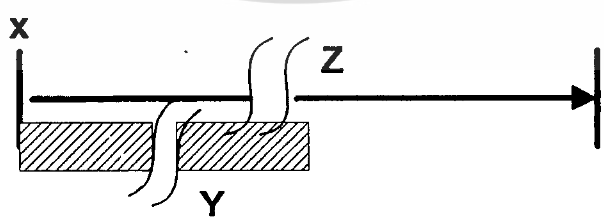


Figure 5.6 schematic of variable time-value type.

In this project, use maximum validation duration scheme because of manage easily and can monitor the realtime performance with fixed interval. This selected mechanism also provide the easy on network traffic management by adjust the realtime interval (Z) if network performance is available.

5.5 System design

On designing system for transmit data between shop floor equipments and back-end processor with fast response, has to consider patten of connection and quantity of data flow in system. There are many equipments and components in shop floor of manufacturing which generate raw data every 5 seconds. Data flow and work load of this data was design to be managed by 4 Netware servers. Each servers handle equipments account seperately by main groups. Data and information of each group has to be managed seperately to increase access time by users or front-end applications.

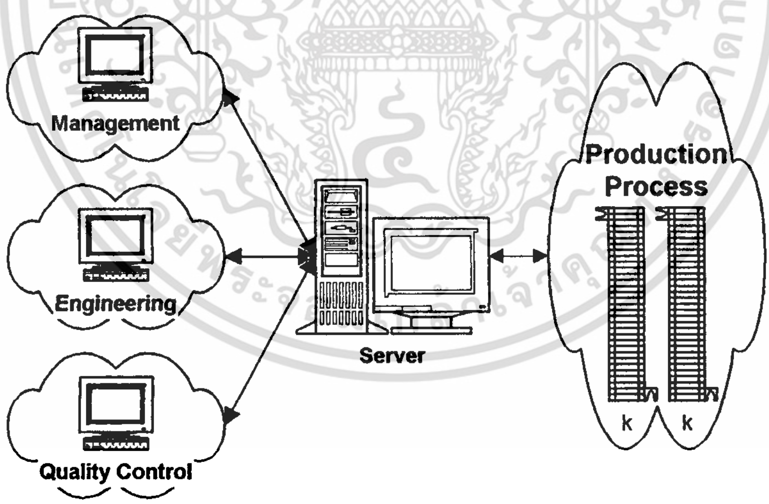


Figure 5.7 connnection of information from shop floor to office level

This project use 2 types of data store to increase access time consider by pattern and format of final information required by users or front-end applications.

1. Information which need selection or query with condition, grouping, ordering will be kept in relational database (RDBMS) which has features support above requirements.

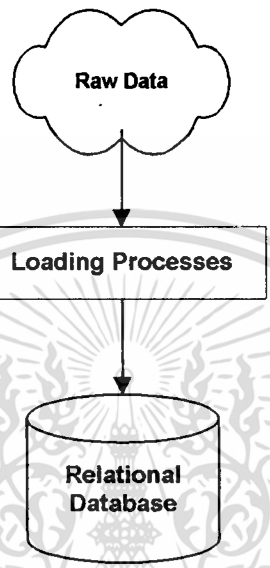


Figure 5.8 data processing for relational database

2. Information which need end result on unit information such as online status of equipment which do not need query feature as relational database will be kept in plain text format with naming convention of filename to support equipment type and date time of data. This plain text format was designed to support fast access of information by front-end applications.

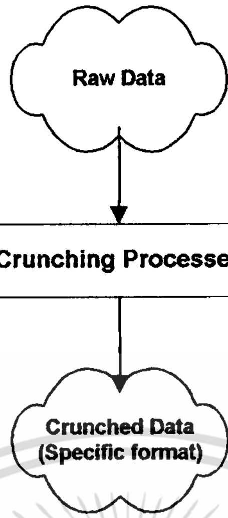


Figure 5.9 processing for plain text database

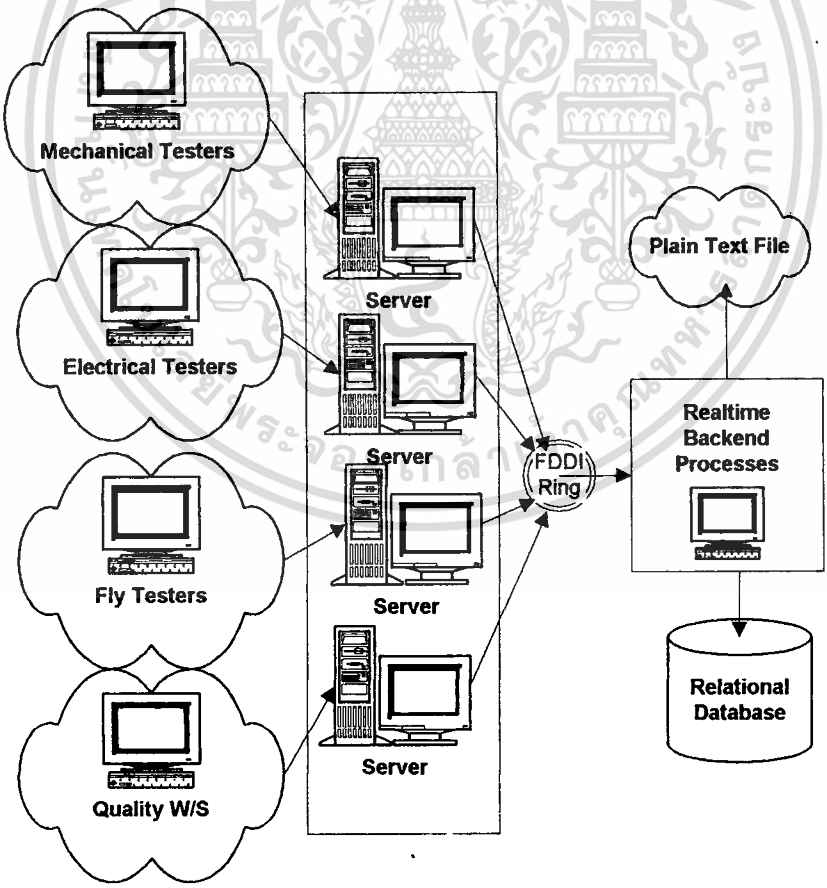


Figure 5.10 seperated data flow of 4 main groups on network through back-end processors

There are 4 main groups of data concerned to this realtime system. Each of this data was designed to be processed seperately by each back-end processor. PCs (personal computer) was setup and programmed to work with assigned routine. Each back-end processor implemented realtime scheme to sleep and invoked on every assigned time value.

With data flow from these 4 main groups of equipments is around 2.8 million records per day, all 4 PCs with specific program has to manage its routine within assigned time value. On design process, 3 factors was considered to determine time interval for each loop.

1. Quantity of input data.
2. Execute time of each cycle.
3. Response time which user need.

Finalize time interval of system is the 10 minutes with acceptance from users. Raw data of 4 main groups was process and manipulate to generate final or crunched information for front-end process. This results of processing were posted into network space for accessable from users and other fron-end applications.

5.6 System architecture

The real world of factory in manufacturing comprise of many production cells. On each cell comprise of many operations which operate routine function by operators. Some oparation may has many operators depend on how much work on that operation. Data entry panel (DNP) was put to each operator's work-bench for collecting data.

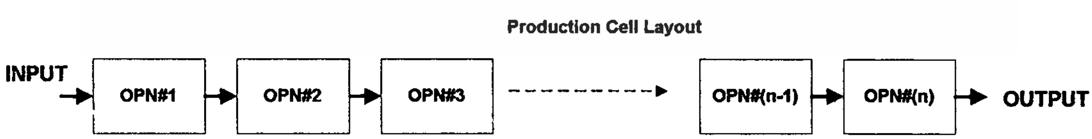


Figure 5.11 Production cell layout

Transaction was collected via the DNPs and pool to host PCs by data terminal system (DTS). This machine has specific program to pooling data from DNPs in realtime and summarize primary report for production cell monitoring. Data from DNPs was encrypted to binary format to protect intervention from unauthorized person. Recovery mechanism was designed by connect power suppply at each DTSs which supply this power to all DNPs in case of power down. All of DNPs has 64K of RAM to store data while waiting for pooling loop of DTS machine to get data. DTS host will looping to gather data from DNPs every 2 seconds and generate realtime primary report for its production cell every 5 minutes. This host send all records of data to network drive which were mounted on Netware4.11 to save realtime data for processing by back-end process of realtime system.

All of DTSs host has account on Netware file server with 10Mb of space. Its realtime data which comprise of 14 fields of data.

MACHINE_ID,TST_DATE,TST_TIME,EMPLOYEE_NUMBER,MODEL, OPN_NAME,CELL,LOT,TAB,SHIFT,PERIOD,PASS,FAIL,DEFECT_CODE, LOCATION

These data was provided by DTSs program into delimited format for loading into database and manupulating by back-end processes. Data of each DNP and operation has its attributes to present all performances of operation. The attributes that interest in this project is the input, output, yield and defect.

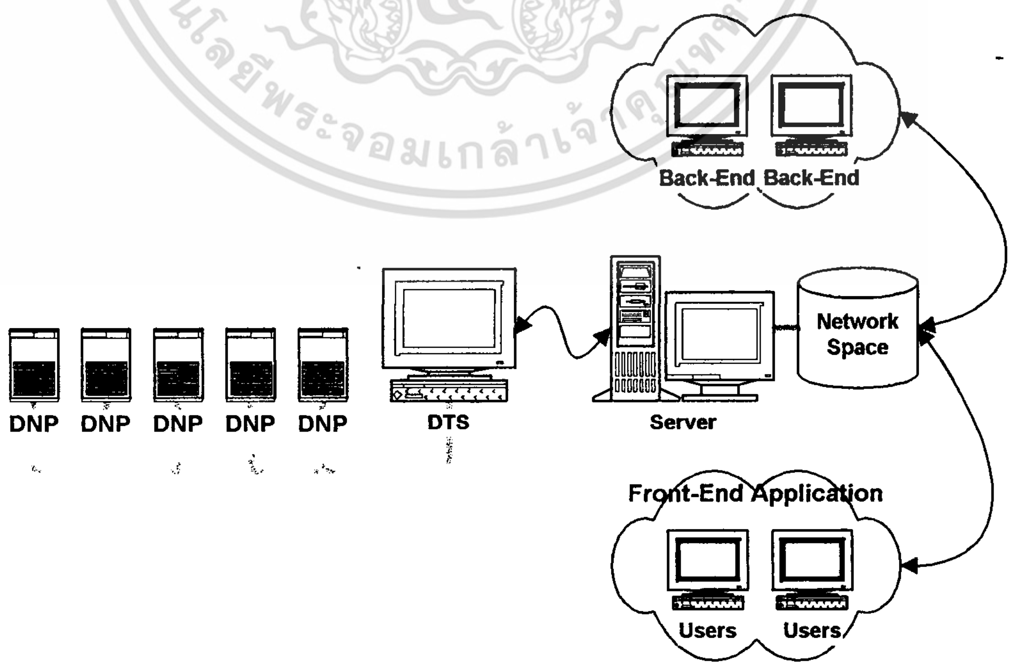


Figure 5.12 schematic of data flow from shop floor to back-end and front-end users

On each production cell has 1 host DTS for handle data pooling from all 50 DNPs. These DTSs were connected to network on same server. Figure 5.13 present the connection of all production cell data into network.

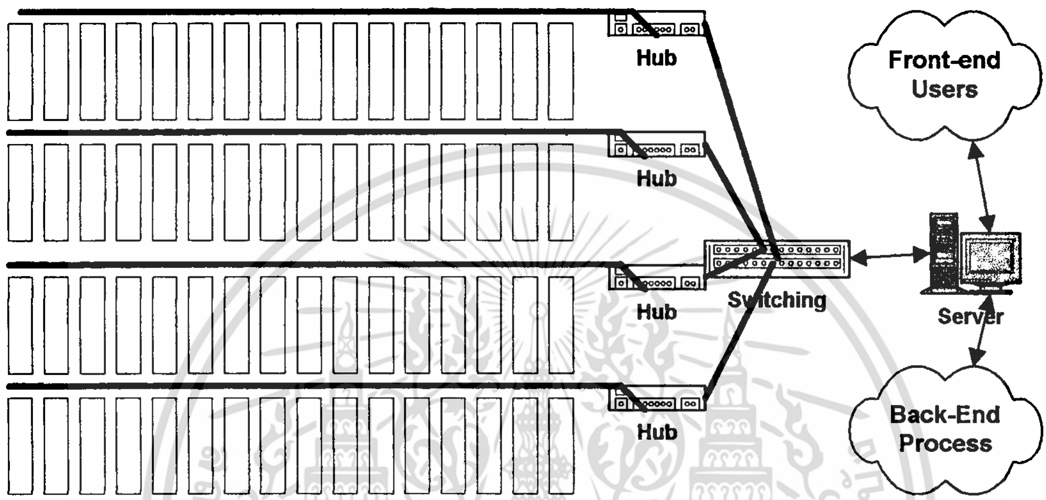


Figure 5.13 Connection of all DTSs at each production cell to enterprise network system

5.7 System development

The development of realtime information system for manufacturing, has some requirements from users to bundle the reporting system into this realtime system. All modules which investigated from users requirements and old batch system are organized as hierarchy in Figure 5.x.

The level of tree was reduce to three level for more efficient memory management and reduce complexity of modules in system. Each modules was designed and developed on user interface by Visual Basic 5 which provide visual programming with objects's properties. Objects in modules will interact with each other by event of object which design and assign in coding program.

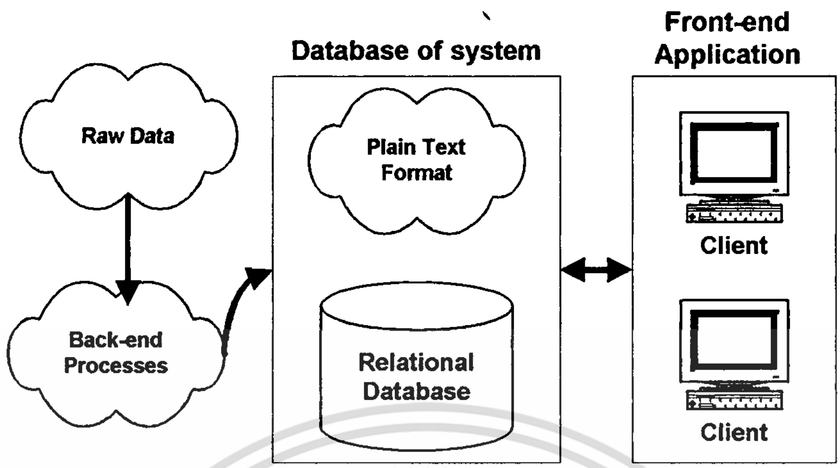


Figure 5.14 Input and output of Back-End processes

Raw data which store in network space was access by back-end process to get and process appropriated function for each group of data. The results of these processing will be kept in 2 types of database format in network space for access by other users or front-end application. System was designed on conflict handling by file replacing mechanism which process information on each period of time and replace the old information after end of processing. This will reduce conflict of file open sharing in some level. File opening conflict was designed to reduce problem by use the file open feature of programming language which Visual Basic support openning file in sharing mode.

Backend-Process

Example function of backend process as show in Figure 5.15. This Back-End#1 connected to network and use account on network which as access rights to data path of raw data.

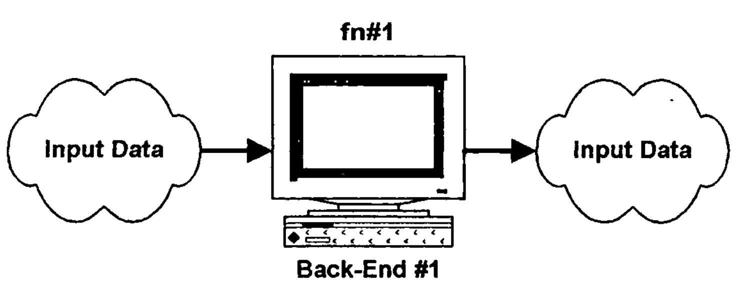


Figure 5.15 Simple data flow through Back-End process

Network drive for this Back-End account will be mapped to appropriated path on many servers for data transfer and processing. Figure 5.16 show access path of Back-End#1 account on Netware servers.

```
Drives A,B,C,D,E map to a local disk.
Drive F: = BKKEIS\DEVELOPER:DEVELOP \BACKEND.JR4
Drive I: = BKKEIS_APPLI.BKKEIS.: \
Drive J: = BKKDNP\DATANFS: \REPORT\TV
Drive K: = BKKENG\USER: \USERS\DEVELOP\YIELD\TMP
Drive L: = BKKEIS_APPLI.BKKEIS.: \
Drive M: = BKKDNP\USER: \
Drive P: = TTENG_APPLI.TTENG.: \APP\JURAS\MES\JRTV
Drive R: = BKKEIS\USER: \TESTER
    ---- Search Drives ----
S1: = X:. [BKKEIS_SYS.BKKEIS.: \PUBLIC]
S2: = Z:. [BKKEIS_SYS.BKKEIS.: \PUBLIC]
S3: = Y:. [TTENG_SYS.TTENG.: \UTILS]
S4: = C:\NWCLIENT
S5: = C:\DOS
S6: = C:\WINDOWS
S7: = C:\NORTON
S8: = C:\NC5
S9: = C:\
S10: = C:\UTIL
```

Figure 5.16 Network map drive of Back-End #1

Main loop was assigned to run on Back-End#1 which is the loop of DOS batch and run on DOS mode. Many tasks was ordered in the batch to function in each assigned routine. Loop will be determined to start when Task#n: meet the condition of sleep time. Each of task was calculate start and stop time of processing in seconds unit. This time interval for each task will use to adjust realtime interval later. Each task can be describe as below.

Task#1: Get online status of tester from network

This module work to get the online status of electrical tester from network by calling Netware function to read network status (NLIST.EXE). GETONLI.EXE is the in-house C program which call NLIST.EXE inside and generate output of this calling to temporary file (ONLI.LOG). It read this temporary file to get the connection list into array for check with configuration file of location of tester in EISCFG file.

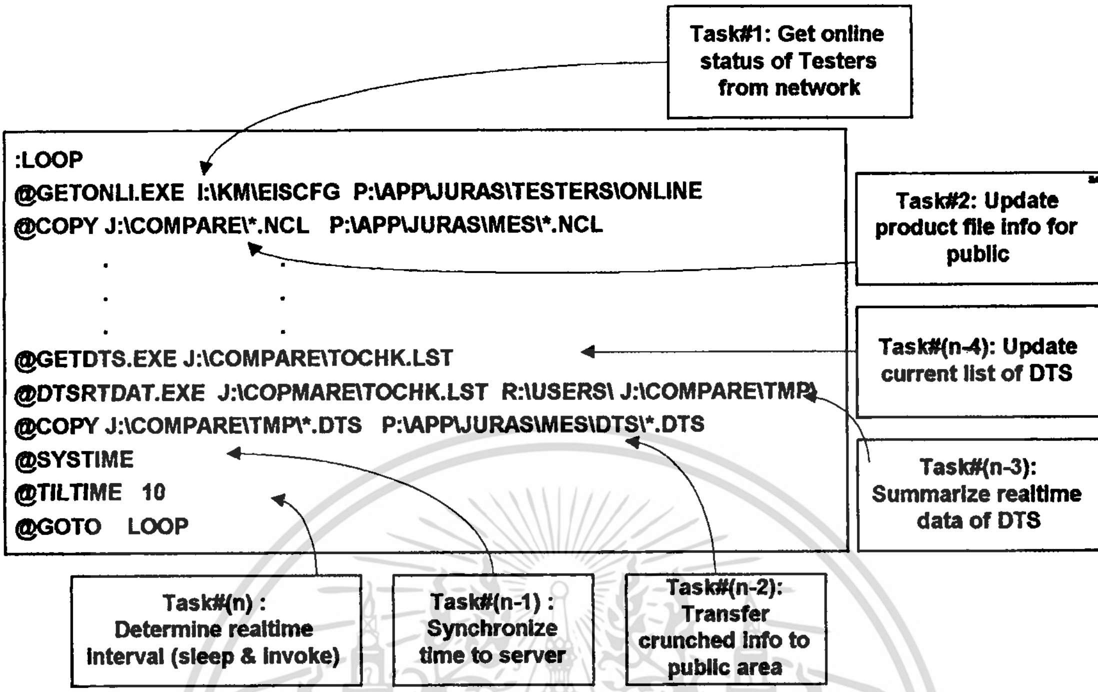


Figure 5.17 Present main loop on Back-End#1 and its tasks

Content of EISCFG file was updated manually by operator to check using testers and actual location of them. Format of EISCFG file as Figure 5.18

TesterName	Location	Position
ECTS89Z	A34	8
ECTS75Z	B34	2
.	.	.
.	.	.
ECTS35Z	D75	5
.	.	.

Figure 5.18 Present details of EISCFG, tester location configuration file

This list will present the tester name, its location of cell in cleanroom which was named orderly. Position of testers identify from top to down of cell. Figure 5.19 present location and position of testers as configured in configuration file, EISCFG.

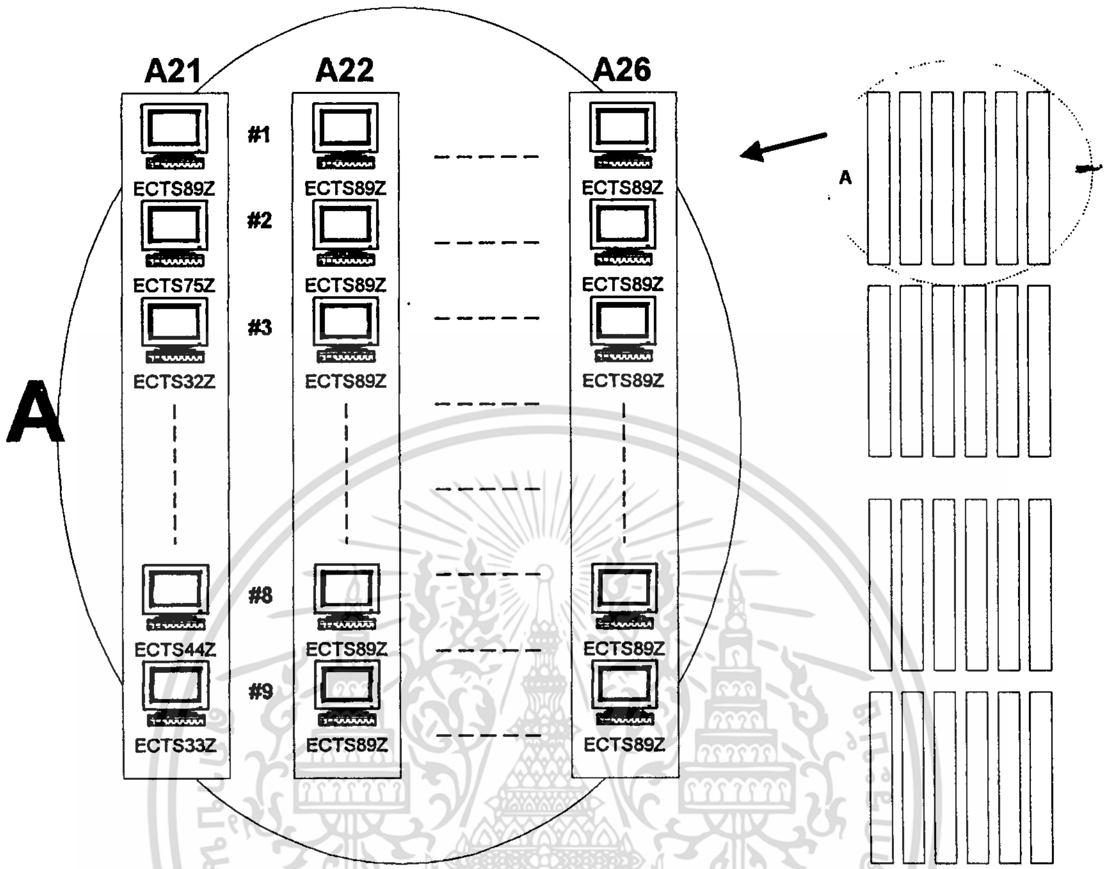


Figure 5.19 Present layout of tester cell in cleanroom.

GETONLI program will compare the list in EISCFG to array of tester which get from online status by calling Netware function. The output log file of Task#1 will be written to P:\APP\JURAS\TESTERS\ONLINE. file which identified as the argument of GETONLI. This file will be accessed by front-end module to present location and position of tester. Figure 5.20 show the output of Task#1.

TesterName	Location	Position	Status
ECTS89Z	A34	8	On
ECTS75Z	B34	2	On
.	.	.	.
.	.	.	.
ECTS35Z	D75	5	Off
.	.	.	.
ECTS37Z	D75	5	On

Figure 5.20 Present output file of Task#1, ONLINE

Task#2: Update product file info for public

Location J:\COMPARE\ is the setup path for modify or change configuration of product which will be build in manufacturing. NCL will be use as extention of file such as BALI.NCL, CUDA4.NCL or BIGBEAR.NCL. These NCLs (New Cell List) file will be increase or decrease depend on how many product build in production. These file use filename convention as product name. Figure 5.21 show content of BALI.NCL file.

```
    Data Path:  J:\COMPARE\_BALI\
     Cell list:  C32,C33,C34,C35,D32,D33,D44
  Target Yield:  98.25
Target Output:  45,000
```

Figure 5.21 Present content of production configuration of BALI

Task#(N-4): Update current list of DTS

This task will work to update in using DTS because some DTS will be off-line for maintenance or some will be move back to laboratory. These list will be up to date every realtime interval to get actual working DTS. DTS name was configured as production cell name for easier on nomenclature. This list will be used for collecting realtime data. Output of this task will be written to J:\COMPARE\TOCHK.LST which contain list of DTS name for task#(n-3) get access to working DTS. This list will be used by other module for tracking purpose also.

Task#(N-3): Summarize reatime data of DTS

This task will read J:\COMPARE\TOCHK.LST file to access working DTS for realtime raw data. Task#3 use mechanism of concatenation DTS name with argument(2) of DTSRTDAT.EXE. By this mechanism, DTSRTDAT know path to access realtime data as Figure 5.22.

```
R:\USERS\A32\REALTIME\
```

Figure 5.22 Concatenated path with DTS name and REALTIME subdirectory

```
x=0;
while(x<MaxDts)
{
        Sprintf(fname,"%s%s\REALTIME\D*.*",argv[1],DtsName[x]);
        Done=findfirst(fname,&ffblk,0);
        while(!done)
                {
                        ....
                        Done=findnext(&ffblk);
                }
        x++;
}
```

Figure 5.23 Present mechanism of concatenated DTSName for access data

DTSRTDAT then use this path to find out realtime data file. At design process, there is commitment to setup file name with convention of time and location of production cell which realtime filename will be.

Dddmhhmm.ccc

which D: means data file

dd: means date (01-31)

m: means encoded month (A:Jan, B:Feb, … ,Dec:L)

hh: means hour of day (00-23)

mm: means minute which file was created (00-59)

ccc: means physical name of production cell (location).

This naming convention will use as the crunched filename for fast access data by identified filename for period of time. Realtime raw data file will collect every 10 minutes as realtime interval. All records was collect in delimited format for easy access by function in VB and C. Data structure of this realtime data as below.

DNP_ID	CHAR(2)
TST_DATE	DATE
EMP_NUMBER	CHAR(6)
PRODUCT_NAME	CHAR(8)
OPN_CODE	CHAR(3)
LOT	CHAR(1)
TAB	CHAR(1)
PERIOD	CHAR(1)
SHIFT	CHAR(2)
CELL	CHAR(2)
PASS	NUMBER(5)
FAIL	NUMBER(5)
DEFECT_CODE	CHAR(4)
LOCATION	CHAR(3)

Totally 14 fields was collected into one record. This one record was key-in by operators from each operation in production cell every 2 minutes.

```
"02","03/10/98 13:11:21"," 68568","_CUDA18 ","120","2","1","1","A","11",0,1,"F12","D72"
"04","03/10/98 13:11:21"," 31199","_CUDA18 ","110","2","1","1","A","11",5,0,"","D72"
"1f","03/10/98 13:11:22"," 49584","_CUDA18 ","040","2","1","1","A","11",32,0,"","D72"
"23","03/10/98 13:11:22"," 70624","_CUDA18 ","145","2","1","1","A","11",5,0,"","D72"
"27","03/10/98 13:11:22"," 60155","_CUDA18 ","140","2","1","1","A","11",5,0,"","D72"
"2d","03/10/98 13:11:22"," 55437","_CUDA18 ","255","2","1","1","A","11",15,0,"","D72"
"33","03/10/98 13:11:22"," 58438","_CUDA18 ","120","2","1","1","A","11",24,0,"","D72"
"3a","03/10/98 13:11:23"," 40436","_CUDA18 ","090","2","1","1","A","11",110,0,"","D72"
"3b","03/10/98 13:11:23"," 82954","_CUDA18 ","090","2","1","1","A","11",5,0,"","D72"
```

Figure 5.24 Present realtime data format of mechanical testing

Delimited format of input data will be processed by DTSRTDAT which developed on C program. Ouput information which users need is in format of text report show as Figure 5.25 and other format of crunched data for graphic report. This report will present performance of production cell in each operation.

Shift: 1(Day)
Cell: A44
Date: 980302

Def	040 L/H	090 W/B	120 G/B	260 SPOT_C	372 100X	380 TRIM_W	390 FOI	450 PACK	SUM	%CUM
	Rej	Rej	Rej	Rej	Rej	Rej	Rej	Rej	Rej	Rej
B2	-	-	-	-	-	-	1	-	1	0.02
F2	-	-	-	2	-	-	3	-	5	0.11
S1	1	-	-	-	-	-	-	-	-	-
S16	1	-	-	-	-	-	1	-	1	0.02
S2	-	-	-	2	-	-	3	-	5	0.11
S23	5	-	-	-	-	-	-	-	-	-
S6	-	-	-	-	-	-	1	-	1	0.02
S7	3	-	-	-	-	-	-	-	-	-
S9	-	-	-	-	-	-	1	-	1	0.02
W12	-	-	-	-	-	-	7	-	7	0.15
TTL	10	-	-	4	-	-	17	-	21	0.45
In	5514	5504	5504	5381	4837	4837	4487	4080		
Out	5504	5504	5504	5377	4837	4837	4470	4080		
F/Y	99.82	100.00	100.00	99.93	100.00	100.00	99.62	100.00	99.55	
							CUM Mech. YLD =		99.55	

Figure 5.25 Present form of final report for Mechanical yield

Task#(n-2): Transfer crunched information to public are

Due to file hadling conflict protection, the source of information which front-end application will open and use have to protect file handling problem. This project use simple mechanism on process and calculate crunched information at temporary path and copy all of them to replace old information at public area which all users access to get for front-end report.

Task#(N-1): Synchronize time to server

Because of system use data and information from many server and process all tasks at local to reduce traffic on network, time syschronize between local PC and servers need to be considered as the key factor. All of realtime information has timestamp for each data group, any mismatch of time will cause non-accurated information in system. This Netware function will get server time and update local time automatically.

Task#(N): Determince realtime interval (Sleep and Invoke)

This mechanism developed on C program to determince time interval for back-end process system to start update and process assigned routines every identified time interval (as argument 1 of TILTIME.EXE). If all batch and routine within main loop finish processing but time interval still not reach maximum value assign as argument 1, program will loop internally to check time until it reach maximum value then release execution from TILTIME.EXE and start all routine by GOTO function of batch.

Those are some example of tasks which were designed and assigned for Back-End#1. All of them work synchronize with server time to process and calculate raw data into final format and some specific format for front-end application.

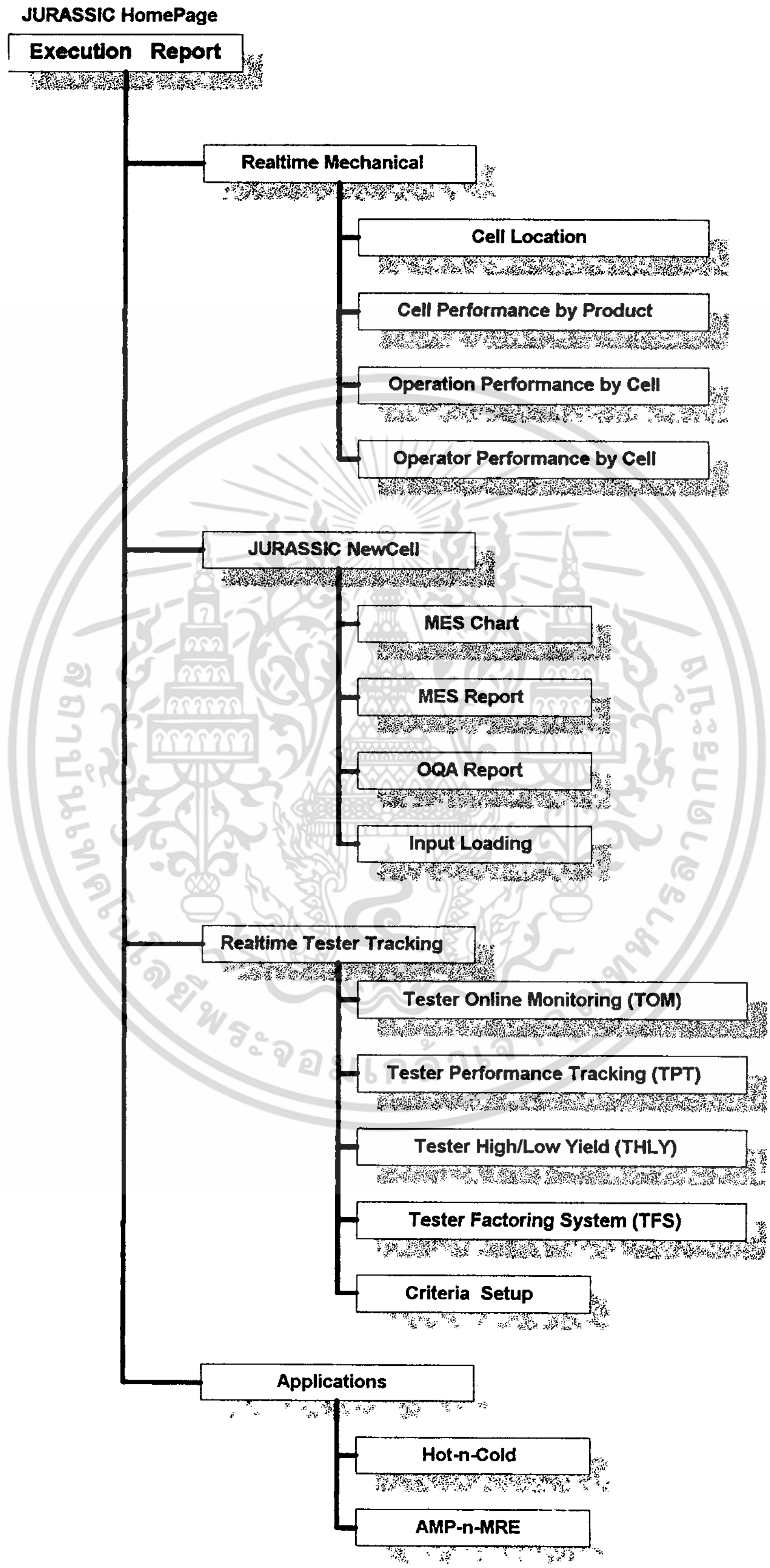


Figure 5.26 modules hierarchy of realtime information system

5.8 System implementation

Realtime program in user interface modules were developed on Visual Basic 5.0 because of easy feature on visual programming and user interface look like object orientation as the real world for users.

This system use 5 PCs running back-end processes which was developed on C program for fast processing and some back-end modules developed on Visual Basic 5.0. All data collect to the system by network connection. Data and informatin was stored in MSAccess database and some information was designed in specific format for implementation of hierarchical decision structure.

System was implement in the real situation of harddisk manufacturing with 2 millions records per day data flow from 80 production cells, 4,000 DNPs. And electrical data flow from 510 electrical testers is around 900,000 records per day.

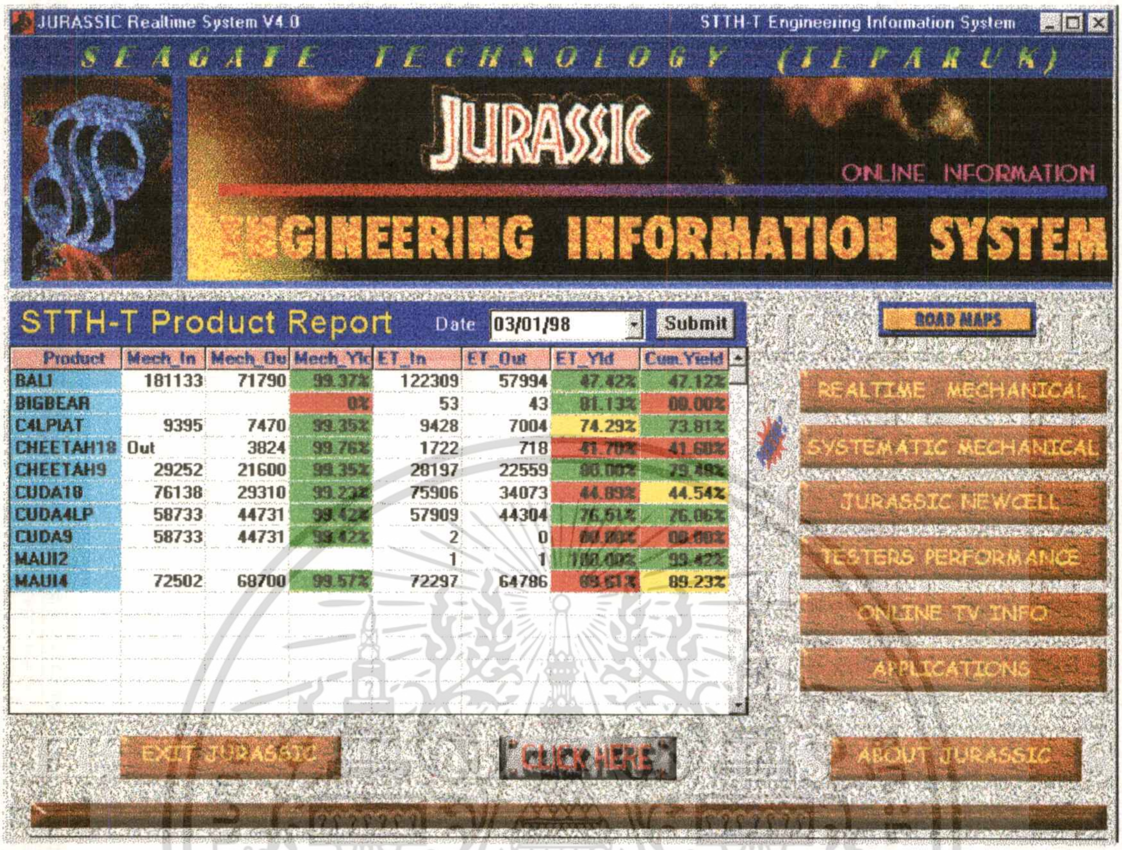


Figure 5.27 User interface first page of developed realtime system

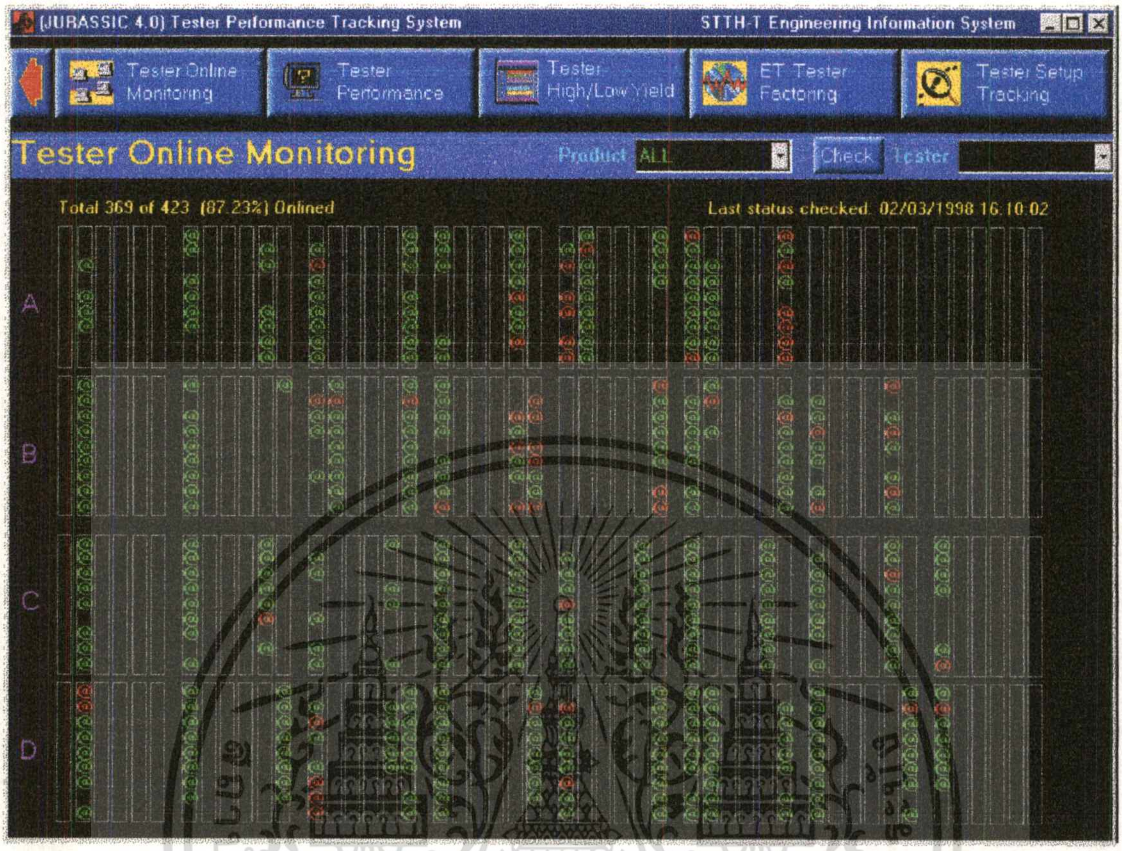


Figure 5.28 Module Online Status Monitoring

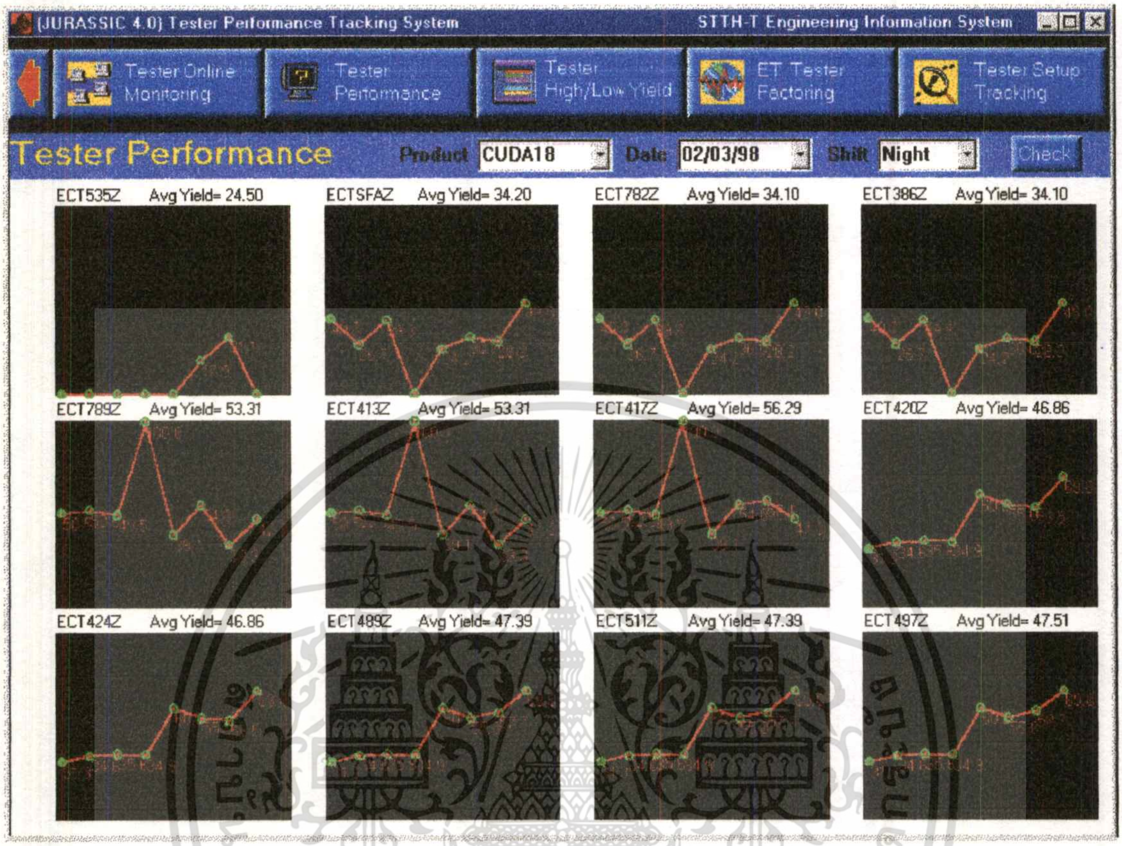
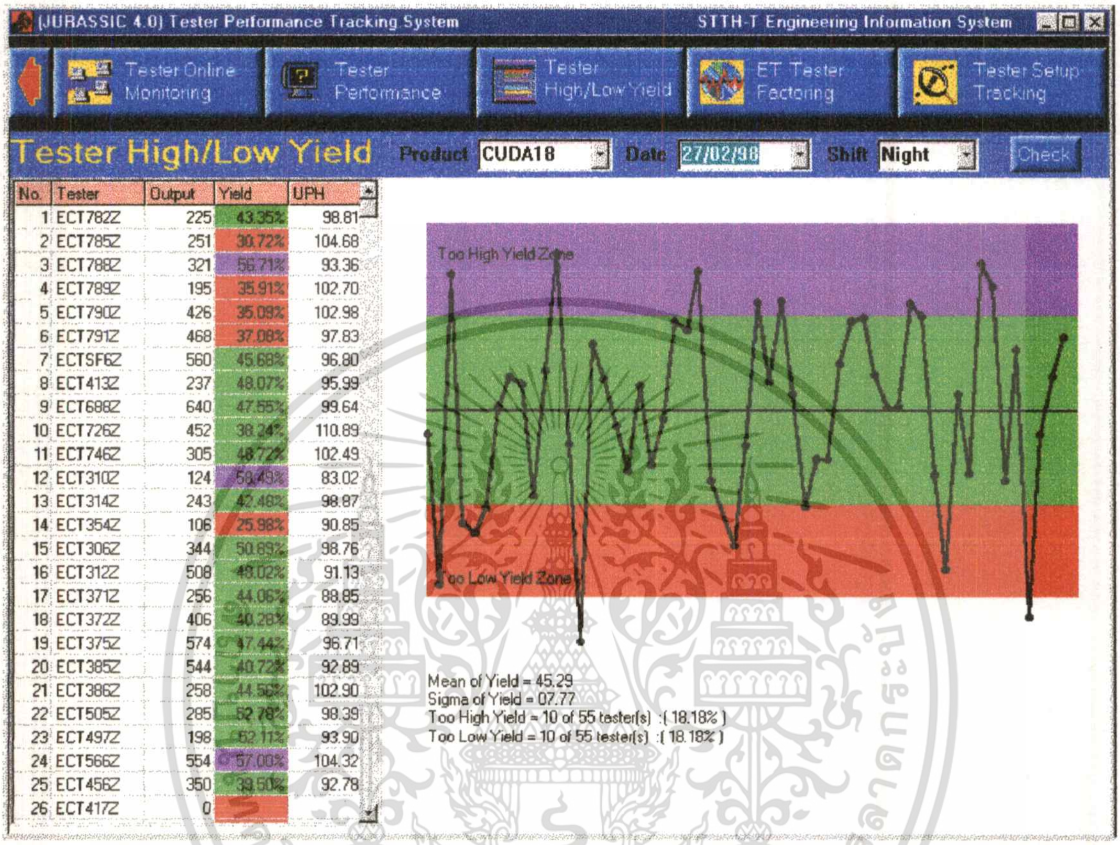


Figure 5.29 Module Tester Performance Tracking



No.	Tester	Output	Yield	UPH
1	ECT782Z	225	43.35%	98.81
2	ECT785Z	251	30.72%	104.68
3	ECT788Z	321	56.71%	93.36
4	ECT789Z	195	35.91%	102.70
5	ECT790Z	426	35.09%	102.98
6	ECT791Z	468	37.08%	97.83
7	ECTSF6Z	560	45.68%	96.80
8	ECT413Z	237	48.07%	95.99
9	ECT688Z	640	47.55%	99.64
10	ECT726Z	452	38.24%	110.89
11	ECT746Z	305	48.72%	102.49
12	ECT310Z	124	58.49%	83.02
13	ECT314Z	243	42.48%	98.87
14	ECT354Z	106	25.98%	90.85
15	ECT306Z	344	50.89%	98.76
16	ECT312Z	508	48.02%	91.13
17	ECT371Z	256	44.06%	88.85
18	ECT372Z	406	40.28%	89.99
19	ECT375Z	574	47.44%	96.71
20	ECT385Z	544	40.72%	92.89
21	ECT386Z	258	44.56%	102.90
22	ECT505Z	285	52.78%	98.39
23	ECT497Z	198	52.11%	93.90
24	ECT566Z	554	57.00%	104.32
25	ECT456Z	350	39.50%	92.78
26	ECT417Z	0		

Figure 5.30 Module Tester High/Low Yield Tracking

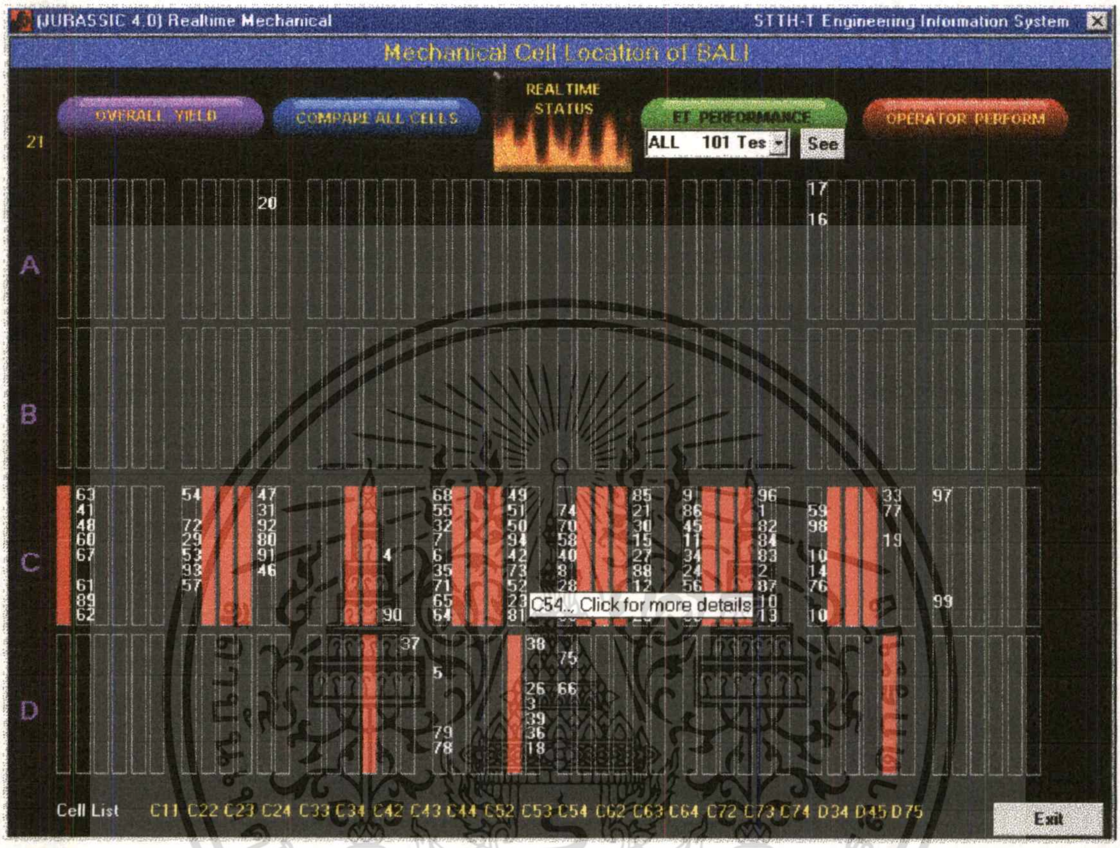
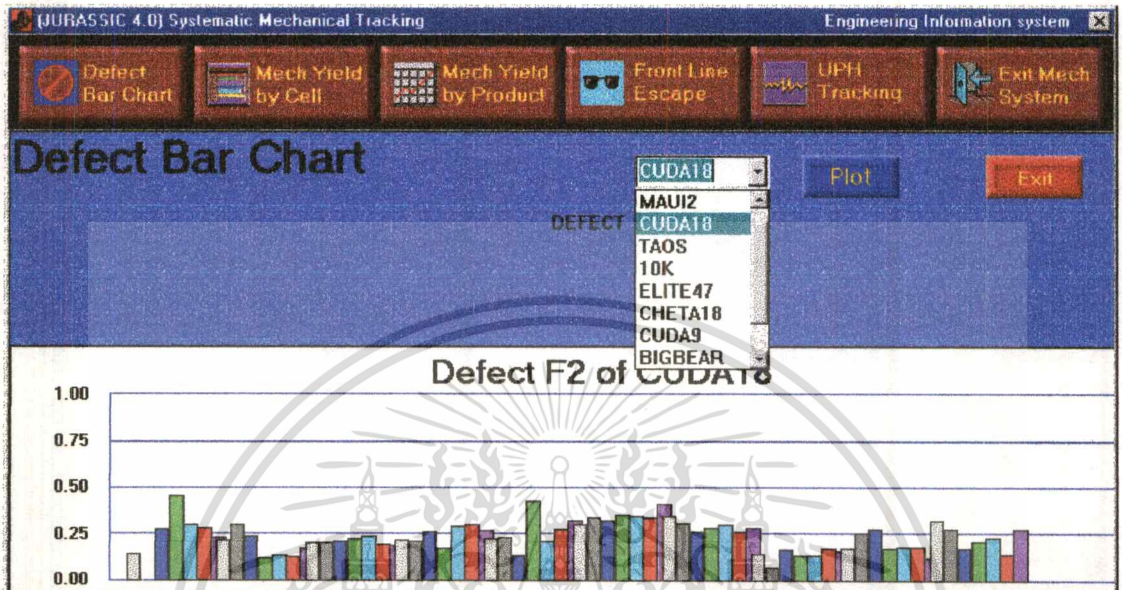


Figure 5.31 Module Realtime Mechanical Tracking



10 Cell(s)		D23	D25	D65	D22	D24	D53	D54	D63	D64	D72	ALL
WW9836		0.26	0.20	0.17	0.21	0.19	0.31	0.17	0.08	0.27	0.22	
WW9837		0.62	0.19	0.15	0.20	0.18	0.29	0.18	0.13	0.28	0.28	
14/03/98		0.14	0.22	0.20	0.21	0.22	0.30	0.34	0.14	0.17	0.32	0.2
15/03/98			0.30	0.20	0.21	0.23	0.33	0.31	0.07	0.26	0.27	0.2
16/03/98		0.28	0.24	0.21	0.26	0.13	0.32	0.26	0.16	0.28	0.17	0.2
17/03/98		0.45	0.12	0.22	0.17	0.43	0.35	0.29	0.13	0.17	0.21	0.2
18/03/98		0.29	0.13	0.23	0.29	0.21	0.34	0.29	0.13	0.18	0.23	0.2
19/03/98		0.28	0.13	0.19	0.29	0.27	0.33	0.26	0.17	0.18	0.14	0.2
20/03/98		0.23	0.17	0.18	0.27	0.32	0.41	0.28	0.16	0.12	0.28	0.2

Figure 5.32 Module Defect Bar Chart reporting

CHAPTER 6

Conclusion

Conclusion

The result of development found that the developed system can provide useful information to manufacturing operation. Not only realtime information which were used by shop floor operators to solve problem but also useful for office person and management level to monitor their concerned functions. This realtime system can track the availability of equipments, status of them and operations. With the GUI interface, user no need computer skill to use this system and with Client/Server architecture, all users in organization can access this useful information world wide. Benefits from developed system on manufacturing has many aspects.

Data collection/acquisition

Data Entry Panel (DNPs) are wired equipments to network system for transfer all necessary data to database system. This data was processed for specific pattern and information need by users. Raw data which was gathered and collected in database use for monitoring, reporting, planning, scheduling and analysis to improve quality and efficiency of manufacturing.

Product tracking

Information which was processed through back-end processing was used to track all operations of product since loading material for assembly till finish goods. This tracking will let response person know about performance and problem in production processes. Provide visibility to where work is at all times and its disposition status information include; components, lot, serial number, current production conditions and any alarms. The online tracking function create an historical record. This record allows traceability of components and wage of each end product.

Resource allocation and status

System provide information for manage resources including machines, tools, equipments, operators and other entities. With the realtime information from the system, any bottleneck occure in production process can be alerted to concerned person thus allocation of necessary entities for improve througput will be actioned by using this information.

Performance analysis

System provide up to the minute reporting of actual manufacturing operations results along with the comparison to past history and expected results. Performance results include measurements the output and defect of part along the processes, product unit cycle time conformance to schedule and performance standards. The performance/downtime of each machine can be used as a base for an improved preventive maintenance schedule.

Process management

System provides information to monitor production and provides decision support to operators for correcting and improving in-process activities. These activities may be intra-operational and focus specifically on machines or equipment being monitored and controlled as well as inter-operational which is tracking the process from operation to the next. This system include alarm management to make sure factory persons are aware of process change which are outside acceptable tolerances.

Maintenance management

System provides information for tracks and directs the activities to maintain the equipment and tools to ensure their availability for manufacturing and ensure scheduling for periodic or preventive maintenance as well as the response (alarms) to immediate problems. This maintain a history to aid diagnosing problems.

Labour management

As the system provides instance information for front-end users to track their performance, supervisors or operators can solve problem themself by using this realtime

information system. This can reduce skill persons such as technicians or engineer from shop floor process and can allocate them to other important areas or problems.

BIBLIOGRAPHY

Ayer, Steve J. Object-Oriented Client/Server Application Development using ObjectPAL and C++. Singapore:Prentice-Hall, 1995.

Date, C. J. Introduction to Database Systems. Fifth Edition., Addison-Wesley, 1990.

Harrington, John and others. Visual Basic 5 interactive course. CA: Wait Group Press, 1997.

Kim, Kan H. "Object Structures for Real-Time Systems and Simulators". IEEE Computer, Vol. 30, No. 8 (August 1997) pp. 62-70.

Renaud, Paul E. Introduction to Client/Server Systems. New York: John Wiley & Sons, 1996.

Sadashiv, Adiga. Object-oriented Software for Manufacturing Systems. Iowa:Chapman&Hall, 1992.

Thomas, Zane and others. Visual Basic 4 how-to. Corte Madera, CA: Waite Group Press, 1995.

Vaskevitch, David. Client/Server Strategies. CA: IDG Books Worldwide, 1995.

BIOGRAPHY

Author's Name Mr. Kosol Chakanjsilp

Date of Birth 2 December 1967

Place of Birth Burirum

Bachelor Degree Bachelor of Computer Engineering

Graduate Institute King Mongkut's Institute of Technology Ladkrabang

Present Carrier Lead Engineer

Engineering Information System, T.E.

Seagate Technology (Teparuk) Ltd.