The Development and Implementation of a Production Information Collection and Reporting System

by

Haitao Liu

Bachelor of Engineering in Metal Forming, Northeastern University (1997)

Submitted to the Sloan School of Management and the Department of Electrical Engineering and Computer Science in Partial Fulfillment of the Requirements for the Degrees of

Master of Business Administration
and
Master of Science in Electrical Engineering and Computer Science and

In Conjunction with the Leaders for Manufacturing Program at the
Massachusetts Institute of Technology
June 2004

Signature of Author____
Sloan School of Management
Department of Electrical Engineering and Computer Science

May 2004

Certified by____
Arnold Barnett, Thesis Advisor
George Eastman Professor of Management Science

Certified by____
Stephen Graves, Thesis Advisor
Abraham J. Siegel Professor of Management Science and Engineering Systems

Accepted by____
Margaret Andrews, Executive Director of Masters Program
Sloan School of Management

Accepted by____
Arthur Smith, Chairman, Graduate Committee
Department of Electrical Engineering and Computer Science

This page is intentionally left blank

The Development and Implementation of a Production Information Collection and Reporting System

By
Haitao Liu

Submitted to the Sloan School of Management and the Department of Electrical Engineering and Computer Science on May 18, 2004 in partial fulfillment of the Requirements for the Degrees of Master of Business Administration and Master of Science in Electrical Engineering and Computer Science

Abstract

Production information, which includes production counts and line downtime information, is of great importance for automobile assembly plants to diagnose equipment problems and improve line utilization. Outdated information systems at many manufacturing plants are not capable of providing accurate production information in a timely manner. This thesis is a thorough account of an internship project conducted at Ford Motor Company's Kentucky Truck Plant (KTP) which turned a spreadsheet-based production information system into a relational database application, called PICRS.

The first part of the thesis introduces the reader to the background of the internship project, focusing on the importance of accurate and timely production information and the inadequacy of the old system. The second part talks about the development of PICRS. Descriptions about system development approach, software and hardware considerations, database design, and interface design revolve around the low cost, usefulness, and ease-of-use of PICRS. The third part of the thesis recounts the implementation process of PICRS and discusses its benefits and impact on KTP's final assembly area. Lessons and observations on leadership, change management, and corporate culture, as mostly obtained in the implementation process, are also discussed in this part of the thesis.

Thesis Advisor: Arnold Barnett
Title: George Eastman Professor of Management Science

Thesis Advisor: Stephen Graves
Title: Abraham J. Siegel Professor of Management Science and Engineering Systems

Acknowledgements

This thesis is a work made possible by a number of organizations and individuals. First, I would like to thank Ford Motor Company Kentucky Truck Plant and MIT Leaders for Manufacturing Program for sponsoring this internship project. In particular, I would like to acknowledge the valuable instructions and help from the following four gentlemen:

My engineering thesis advisor: Arnold Barnett (MIT)
My management thesis advisor: Stephen Graves (MIT)
My company supervisor: Glenn Whitfield (Ford)
My company advisor: Chris Pereira (Ford)

In addition, I would like to specially thank Chris Delaney, a graduate student majoring in Manufacturing Engineering at the University of Kentucky, with whom I developed PICRS system, a major deliverable of the internship project.

I also owe gratitude to the following individuals who have helped to make all this happen:

Donald Rosenfield, Nancy Yang, and Patty Sullivan at MIT;
Mike Parris, Amy Snider, Ronald Parson, and Will Ware at Ford Motor Company.

I dedicate this thesis to my dear wife Joan.

Table of Contents

List of Figures

List of Tables

This page is intentionally left blank

Chapter One: Introduction

1.1 Industry Background

Troubled by overcapacity and huge healthcare and pension liabilities, the US Big Three automakers (GM, Ford, and DaimlerChrysler's Chrysler arm) have been losing ground to foreign competitors, especially the Japanese Big Three (Toyota, Honda, and Nissan). For the last seven years, Detroit's share of US car and light truck Market has dropped from 72.5% in 1996 to 63.9% in 2003, while that of the Japanese Big Three has risen from 17.9% to 22.6%[1], and the trend is still favoring the Japanese automakers.

America's auto industry has one-fifth more capacity than it needs[2]. However, its ability to downsize is restricted by its labor contracts with the unions. Moreover, retirement now brings its own peril of growing pension burden. Add the healthcare liabilities, and US auto industry is at huge competitive disadvantage to its Japanese competitors.

Import tariff could be a protection to US auto industry if the foreign vehicles were made outside of US. But now the foreign automakers have built and are building more plants in the southern states of America. With waning protection from tariffs, and growing burden in pensions and healthcare liabilities, US automakers have to seek ways to survive the fierce competition from foreign competitors.

America's growing appetite for sport utility vehicles (SUVs), minivans, and pickup trucks seems to be a savior. Now about every other passenger vehicle sold in the US is of one of the three categories. On average, SUVs and trucks are in stronger demand in terms of manufacturing capacity and bring higher margins. In recent years, those two types of vehicles have accounted for most of the Detroit Big Three's profits. Therefore, plants that produce SUVs and light trucks are now of elevated importance to US automakers. Many truck plants are producing to capacity to cater to the market. In this light, productivity is of greater importance in those SUV and truck plants than in other Big Three automobile assembly plants.

The competition from Japanese automakers has also urged the US automakers to modernize their plants and imitate the improvement-driven Japanese manufacturing techniques. Specifically, US car makers are becoming more cost-cautious, more meticulous on quality, and more aggressive on product development. Process improvement is a major means to realize better quality and lower cost. Lean manufacturing and continuous improvement concepts and tools have been widely applied in almost all facets of manufacturing in the US automotive industry. Buzz words such as *Lean Manufacturing*, and *Six Sigma* have become common place in most of the manufacturing plants. In continuous-flow assembly plants, *Theory of Constraints* are eagerly explored, and at some places, well practiced to identify and manage bottlenecks. Now, every automaker claims to have its own version of the famous *Toyota Production System* (TPS). TPS and some of its terms have been associated with flexibility, high quality, and low cost. However, the results of US automakers imitating Japanese techniques have not been uniform. They vary from one maker to another maker and from one plant to another.

This thesis project was done in a major pickup truck assembly plant of Ford Motor Company in such an industry background. The above information can help the readers to understand the bigger macroeconomic background behind the importance of improving productivity at this particular plant.

1.2 Ford Kentucky Truck Plant and Its Final Assembly Area

Ford Motor Company (Ford) is the second largest automaker in the world with 2002 revenue of a little short of $163 billion, and US passenger vehicles market share of roughly 20%. Ford owns eight brands which include three domestic brands: Ford, Mercury, and Lincoln, and five acquired foreign brands: Jaguar, Land Rover, Aston Martin, Volvo, and Mazda. Ford has been experiencing difficulties for the last few years because of the fierce competitions mentioned earlier and a failed strategy by the previous CEO, Jacques Nasser, to transform the company into a global consumer-services company. In 2001, Ford had a net loss of $5.45 billion. The loss dropped to $980 million for year 2002. For the first three quarters of 2003, Ford has gained a net profit of about $1.3 billion. Current CEO Bill Ford's campaign of "Back-to-basics" seems to be working. In 2002, the North America capacity utilization was increased by 10%, while non-product cost was cut by $2 billion. Also in 2002, Ford was up 12% in J.D. Power and Associates initial quality study. All these numbers reflect the company's efforts to cut cost and improve quality, and, as Bill Ford put it, "Focus on the fundamentals" (Ford's strategy to cut off non-core businesses and focus on vehicle making).[3]

Ford F-series trucks have been the best selling vehicles in the US for the past two decades. Since their debut in 1948, Ford has sold 27.5 million F-series trucks. These trucks are the major contributor to Ford's profit for years. Even in the recent years of difficulty, the sales of F-series trucks remains strong and have become more important in company profit contribution. Ford F-series trucks for North America market are built in several locations, including Kansas City Assembly, Norfolk (Virginia) Assembly, Ontario Truck, and Kentucky Truck. Norfolk and Kansas City plants are now building the redesigned F-150 on their revamped flexible lines. In addition, a newly-built plant in Ford's heritage Rouge Complex will start assembling the all-new F-150 trucks in mid-2004. Ford Kentucky Truck Plant (KTP), located in Louisville, Kentucky, is the only plant building super-duty F-series trucks for North American market.

KTP is a 4.6 million square-foot facility, which employs roughly 6000 salaried and hourly workers and produced 378,303 vehicles in the year of 2002. The models currently built at KTP include Ford Excursion SUV and Super Duty F-Series trucks (F-250, F-350, F-450, and F-550). Currently, KTP is operating on three shifts. The whole plant is in full production about 20 hours a day, except for a 24-hour shutdown starting at Saturday night for maintenance purposes.

KTP is made up of five main production areas, which are, in the order of material flow, Stamping Area, Body Shop, Paint Shop, Trim Area, and the Final Assembly Area. The Final Assembly Area consists of two parallel assemblies. Each assembly is further divided into two lines. The first line is called frame line, which starts with a bare truck frame and ends with engine, transmission, tires, and other accessories on it. The second line is called chassis line, where cabs and boxes are decked onto the frames and all the other finishing jobs, such as steering wheel install and fuel fill, are done. Each line is moved by a single master conveyor. There is a decoupling buffer between each frame line and its successive chassis line. There is no crossover between the two frame-chassis lines. Besides the four main assembly lines, there are engine line and tire room which feed into the frame and chassis lines. Immediately after the chassis lines, there are five toe-in and seven rolls testing stations, which are, in the strict sense, the very final stage of truck production at KTP. Currently, the final assembly area is the bottleneck of the whole plant. In the final area, the chassis lines are having more downtimes than frame lines. Downtime data indicates that the balancing between frame and chassis is also a very important issue.

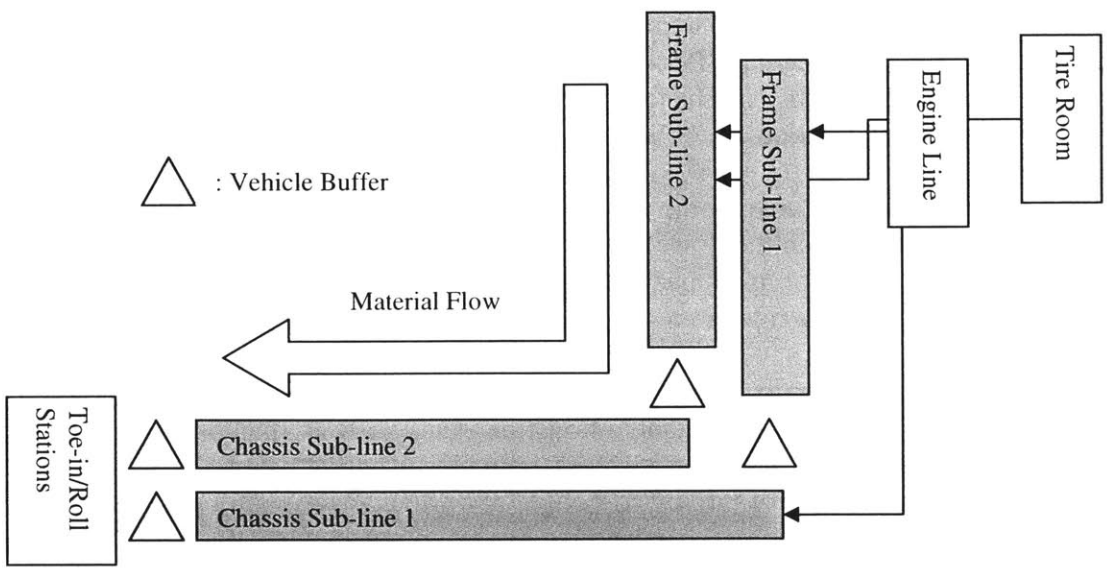


Figure 1: A sketchy map of KTP final assembly area

1.3 The Importance of Timely and Accurate Downtime Information

Currently, the demand for F-Series Super Duty trucks exceeds the capacity of KTP. In this light, increasing throughput is understandably the most pressing issue at KTP. To improve the throughput of a continuous process, one must know where the bottleneck is, and come up with feasible ideas to increase the capacity of the bottleneck. However, in a dynamic manufacturing environment, such as an automobile assembly, the location of bottleneck can change over time. Even at the same time, there might exist multiple bottlenecks due to the decoupling points.

Actions taken to smooth production and increase capacity have to be data-driven. As the assembly line speed grows increasingly fast, the requirement on the timeliness of the process data is becoming increasingly high.

Downtime information is a very important kind of production information. It is useful in the following ways:

- It is necessary in identifying constraints;
- It provides information to differentiate special causes from common causes of problems;
- It provides information on the conditions of equipment.

Since downtime data is so important, all manufacturing organizations want to collect and use it in some way. However, the collection and analysis of the downtime data is not always an easy job. In the KTP final area, the supervisor of each shift was required by the area managers to jot down the production number and downtime information of his or her line each hour of production. At the end of the shift the supervisor needed to come up with a consolidated report to the area managers based on the downtime information he or she jotted down during the day. Periodically (every week and month), certain engineers needed to create a report on Overall Equipment Effectiveness (OEE) based on the downtime data for that period.

To make all this possible, some engineers, years ago, devised a set of spreadsheets for the supervisors to put in the downtime and production count information and generate the reports. What the supervisors needed to do was to fill out one separate Excel file for every shift. All lines in the final area share the same files through a shared drive.

It might be surprising to know that, in an advanced manufacturing facility such as KTP's final assembly area, important production data, such as that on downtime, had been manually input to simple spreadsheets until this August, when the new database system was implemented. By talking to people at various organizations, the author has learned that Excel Spreadsheet is indeed a very common means of storing important process data even in large organizations with strong IT capabilities.

Spreadsheets can meet the basic requirement of storing data, but there are a number of serious disadvantages associated with it. The following are the disadvantages of using spreadsheet to manage data at KTP's final area.

- A spreadsheet can be a repository of data, but can hardly be a database in a strict sense, because there is no easy way to input, retrieve, and query large amount of data to and from a spreadsheet.
- It was time consuming to input data into the spreadsheets. It was even more so to consolidate the data to generate a regular report on line downtimes. Ad hoc reports on line downtimes were out of the question, because almost nobody in a

busy production environment wanted to spend the time to go through the spreadsheets to come up with some information even though it might be useful.

- Manual input was an error-prone process. There was no mechanism to prevent or reduce the accidental input mistake. The descriptive short sentences about downtime reasons written by line supervisors were of no fixed format, which made it impossible to do any automatic sorting and compiling of the data.
- It was hard to organize and make use of the large amount of Excel files in existence. And the number of files was still growing everyday. So, most of the time those "data" files just sat there untouched by anybody.

1.4 Summary of Project

The thesis project originated with KTP's need to keep track of the large amount of useful downtime information in the final assembly area in order to improve KTP's overall throughput measured by average job per hour (JPH). The initial hope was to obtain the information by implementing a piece of commercial-off-the-shelf software named VisualPlant. Unfortunately, some unsolvable financial issues caused the abortion of the purchase of the software. However, the aborted attempt to use this commercial software only served to create an opportunity for the author of this thesis to build a system that could perform some of the needed functions of VisualPlant, and even can overcome some of its limitations. The in-house built system turned out to be a success, and the descriptions of its development, implementation, and benefits are the centerpiece of this thesis.

The author spent his first month of the internship getting familiar with KTP's manufacturing processes, organization structure, and informal networks. During that month, the author also investigated alternative potential solutions to the problem, and gradually defined the goal, scope, and approach of the project. The second month of the internship saw an intense period of requirements definition, and system design and development. In this month, the author spent the bulk of the time interviewing future users about the requirements, and mapping all the major equipment on the chassis lines in order to compile a comprehensive downtime reason list. The rest of the time was devoted to the design and development of the system, much time being consumed by programming and debugging.

The system was implemented onto the chassis lines and test run by A crew only on August 11th, 2003. After the successful implementation of the system on the chassis lines, the system was further rolled out onto frame lines on September 2nd, and engine lines on September 30th. During the implementation processes, the system itself was also constantly being debugged, and improved. Besides the five production lines, maintenance personnel, industrial engineers, and production managers also got involved in the system development process. Some functions were specially designed for these three groups of user to interact with the system.

In the last two month of the internship, the author spent most of the time analyzing the data collected by the system, and assessing the impact of the new system, while still improving the system based on user feedback. Some efforts were devoted to the improvement of a simulation model on Frame 2- Chassis 2 buffer. Some insights were gained as to how the characterization of the data would affect the accuracy of the model, but due to the limitation of the data, the simulation model remained largely the same.

Toward the end of the internship project, a survey was handed out to all the users of the downtime collection and reporting system. The user feedbacks were compiled and are discussed in Chapter 5 of this paper.

Chapter Two: System Development Part I—Development Approach

The stringent financial situation of Ford Motor Company and the experimental nature of the project determine the principal guideline of system development, which is to achieve maximum effectiveness with minimum cost. Effectiveness can be measured by saved time, improved productivity and quality. Cost includes labor and expenses. The system development was conducted under this guideline, and the following sections describe several major issues in the system development process.

2.1 User Requirements

The first task in developing an information system is to gather user requirements. For this project of developing the Production *Information Collection and Reporting System* (referred to as PICRS hereafter), we have two avenues to collect the user requirements. Firstly, we can examine the old Excel-based system to find what functions the old system can provide. We deem these functions as the baseline requirements for the new system. Secondly, we can interview the users to find the additional functions that they desire. The users include line supervisors, area managers, industrial engineers, maintenance engineers, and material handling personnel.

A study of the old Excel-based system provides the first list of system requirements. These requirements encompass all the functions that the old system can provide. These requirements, summarized as follows, are deemed as baseline functions of the new system:

- PICRS should provide an easy-to-use means for the supervisors to enter production counts and downtime information.
- PICRS should be able to generate daily summary downtime report, daily detailed report, and the job number report.
- There should be a convenient way for supervisors to send hourly downtime information to managers through electronic pages.
- Managers should have access to PICRS reports from their personal computers.

We then interviewed users to gather additional requirements, which they would like to have but the old system could not provide. These additional requirements can be summarized into the following list:

- All the users should have their own customized user interface, and should be able to interact with the system easily.
- Reports about downtime information of any time range should be easily generated.
- Reports about the history of specific downtime reason should be easily available.
- The system should be able to communicate with some existing Excel templates for the purpose of reporting and data analyzing.
- Maintenance personnel should have their own input form for "maintenance actions" and should be able to view the reports with maintenance comments.

- This new system should cover an additional production line—the engine line—which the old system did not cover.

2.2 System Development Approach

For any information system development project, only systematic approaches can possibly avoid delay and deliver required functions. There are hundreds of information systems development methodologies (ISDM) in existence. Iivari et al[4] summarizes IS development approaches into five major groups, which include, among others, structured approaches, and information modeling. Among structured approaches, the waterfall model, rapid application development (RAD), the spiral model, and rapid prototyping are the most popular.

We adopt the evolutionary prototyping approach, one of the prototyping techniques, to develop the system. Yao-Chin Huang, in his paper "Prototyping: Throwaway or Evolution"[5] enumerates the advantages of evolutionary prototyping as: the improvement of user communication, the reduction of the cost, the decrement of the system development cycle, the increment of the final system's quality, the completion of the prototype being that of the system, and the prototype being used as a training facility.

We built a prototype system in about two weeks, and then implemented the prototype in two of the five lines in KTP's final area. We then upgraded the prototype by adding more functions without changing the basic structure, and expanded the implementation of the system into the rest of the area. Our observations agree with Huang's assertions. In particular, the prototyping approach, by allowing users to interact with the system early on, greatly helped us to acquire further user input two weeks into the development stage. The first prototype was also critical to earn the buy-in from the users at a rather early stage.

2.3 Hardware and Software Considerations

2.3.1 Hardware consideration

It takes both hardware and software to build an information system. The development of PICRS is no exception to this rule. However in this specific case, there was relatively little concern about hardware selection for two reasons. First, purchase of expensive hardware is out of the question because of the very limited budget of the project. Second, the existing hardware has already met the basic requirements of a small-scale, manual system. Bearing these in mind, we built a system using only hardware that was already available at KTP.

The hardware structure is a very simple one. It currently consists of 27 workstations and personal computers (the number of users has been increasing, and each user adds one workstation or PC to the system) all of which have access to a shared drive through

TCP/IP protocol. The users write to and fetch information from the shared drive. The details of the network can be summarized in Table 1.

Location	# of PC or Work Station	Function
Toe-in and Roll Stations	2 Work Stations	Toe-in & Rolls Data Input
Chassis Line 1	2 Work Stations	Chassis 1 Data Input
Chassis Line 2	2 Work Stations	Chassis 2 Data Input
Chassis Area Offices	3 PCs	Report Generation and Review
Frame Line 1	1 Work Station	Frame 1 Data Input
Frame Line 2	1 Work Station	Frame 2 Data Input
Engine Line	1 Work Station	Engine Data Input
Frame Area Offices	2 PCs	Report Generation and Review
Maintenance Offices	2 PCs	1) Maintenance Data Input 2) Report Generation and Review
Plant Manager Office	1 PC	Report Generation and Review
Engineer Cubicles	5 PCs	1) Report Generation and Review 2) Data Analysis
Developer Cubicles	2 PCs	System development

Table 1: System Hardware Overview

Each PC or work station of the network is equipped with an Intel Pentium 3 or 4 CPU, at least 128 MB RAM, and has a local hard disk of at least 20 GB capacity. The shared drive has the size of 200 GB, and is accessible to each computer in the network with each user's account through Microsoft Windows network. Each computer uses either Windows 2000 or Windows XP operating system and has the full suite of either MS Office 2000 or MS Office XP.

2.3.2 Software Consideration

The first question to ask in software consideration is whether to purchase the software off the shelf or to build it in house. Normally, the pros and cons of commercial-off-the-shelf (COTS) and built-in-house software can be summarized in the following table.

Buy v.s. Build	Pros	Cons
Buy	No development cost, save time, some guaranteed functions	Not always tailored to the user's needs; maintenance may be a problem; no control of the source code, hard to upgrade
Build	Customized to user's need; easy maintenance; can be more integrated to existing systems; have control over source code	Make take long and costly development process; highly risky outcome

Table 2: Software consideration, Buy v.s. Build[6]

In this specific case, things are a little different. COTS production software, such as VisualPlant is expensive and takes longer time and more effort to implement, whereas an in-house built application incurs almost no cost and takes much shorter time to develop and implement. A clear advantage of VisualPlant is its powerful functionalities. However, at the same time, there are some functions, such as manual input of specific downtime reasons, that VisualPlant can not do while an in-house built application is easy to accomplish. Ford KTP actually attempted to purchase and implement VisualPlant. The plan eventually collapsed due to financial reasons. Thus, to build in house was actually the only choice.

There were two sets of software solutions in consideration at the start of the project. The specifics of both solutions are summarized in table 2.

	Solution 1	**Solution 2**
Back End Software	MS SQL Sever or Oracle	MS Access 2000
Front End Software	ASP Web Programming	MS Access 2000 + VBA programming

Table 3: Software solution options

The initial software consideration was affected by our experience with a failed web-based application attempting to achieve similar functions as PICRS provides. The application, named Plant Floor Metrics Reporting system (PFMR), was developed at Ford's headquarter at Dearborn Michigan. Implementation of PFMR was tried at KTP final assembly area but failed due to several serious limitations and flaws.

- The application was not tailored to the needs of KTP final assembly area. Several much needed functions were not provided by PFMR.
- PFMR was cumbersome and time-consuming to use.
- Several limitations, such as the inability to take a fraction of unit loss, hindered the system's capability to provide accurate information.
- The users had no right to modify or improve the system.

We obtained accounts with PFMR and played with it on our PCs for about a week, and found that the limitations were indeed too huge to circumvent. The application's user unfriendliness was especially notable. At the start of each session, the user needs to take about 20 seconds to log into the system. Since it is web-based, each data entry takes a couple of seconds for the packets to be sent to the server. Moreover, after a certain time of idleness, the user will be automatically logged out; and to resume use of the application, the user has to take time to re-login. All these "features" make the system cumbersome to use. We nevertheless recognized that all the limitations are surmountable, except the slow data entry. The time-consuming data input is inherent with the web-based structure. Whereas data transmission speed is a critical concern in a manufacturing environment, data communication means provided by web browsers limit the data transmission speed. The line supervisors just can not afford to wait a couple of seconds after each mouse click. Our experience with the cumbersomeness of the web-based PFMR promoted our doubt about the feasibility of a web-based structure for PICRS.

Moreover, the MS SQL Sever + ASP solution would incur considerable cost in software purchase, while the software required by MS Access + VBA solution is already available on every workstation and PC.

The pros and cons of the two software solutions are summarized in table 4. We chose the MS Access + VBA solution for its cost-saving, speed, and simplicity.

Software Solutions	**Pros**	**Cons**
MS Access 2000 + VBA	No need to purchase additional software, save cost; fast-speed data entry and retrieval	Not as powerful as Oracle or SQL Sever Database; poor portability
SQL Sever/Oracle + ASP	More powerful database; high portability	Need to buy costly software; Slow data entry/Retrieval; longer development cycle

Table 4: Pros and cons of two software solutions

Chapter Three: System Development Part II—System Design

3.1 Database Design

A database is a collection of information that is organized so that it can easily be accessed managed, and updated. Among the various organizational approaches to databases, the relational structure is the most prevalent one. A relational database is a collection of data items organized as a set of formally-described tables from which data can be accessed or reassembled in many different ways without having to reorganize the database tables[7].

A relational database, compared to databases using other organizational approaches, such as distributed databases and object-oriented databases, is relatively easy to create, access, and extend. A relational database is often represented as a set of tables containing data of predefined categories. Each column of the table represents a certain category of data, for example, the telephone numbers of all employees of a company; and each row is a unique incidence of the data for the categories defined by the columns. The categories are also called *fields*, and a row equals a *record*.

Data tables in a relational database are related to each other by *primary* and *foreign Keys*. A primary key is a column or columns in a table whose values uniquely identify each row in a table. A foreign key is a column or columns whose values are the same as the primary key of another table. The relationship is made between two relational tables by matching the values of the foreign key in one table with the values of the primary key in another.[8]

3.1.1 Entity Relationship Model

Data modeling is one of the most critical development processes in database design. Data model is the conceptual representation of the data structures that are required by a database. For relational databases, *entity relationship (ER) model* is the widely used way to unify the network and relational database views.

The ER model was originally proposed by Chen in 1976[9]. To simply put, ER model is a conceptual model that represents the real world as entities and relationships. The entity relationship diagram is a basic tool to visually depict the data objects and their relationships.

Below is the ER for the database of PICRS.

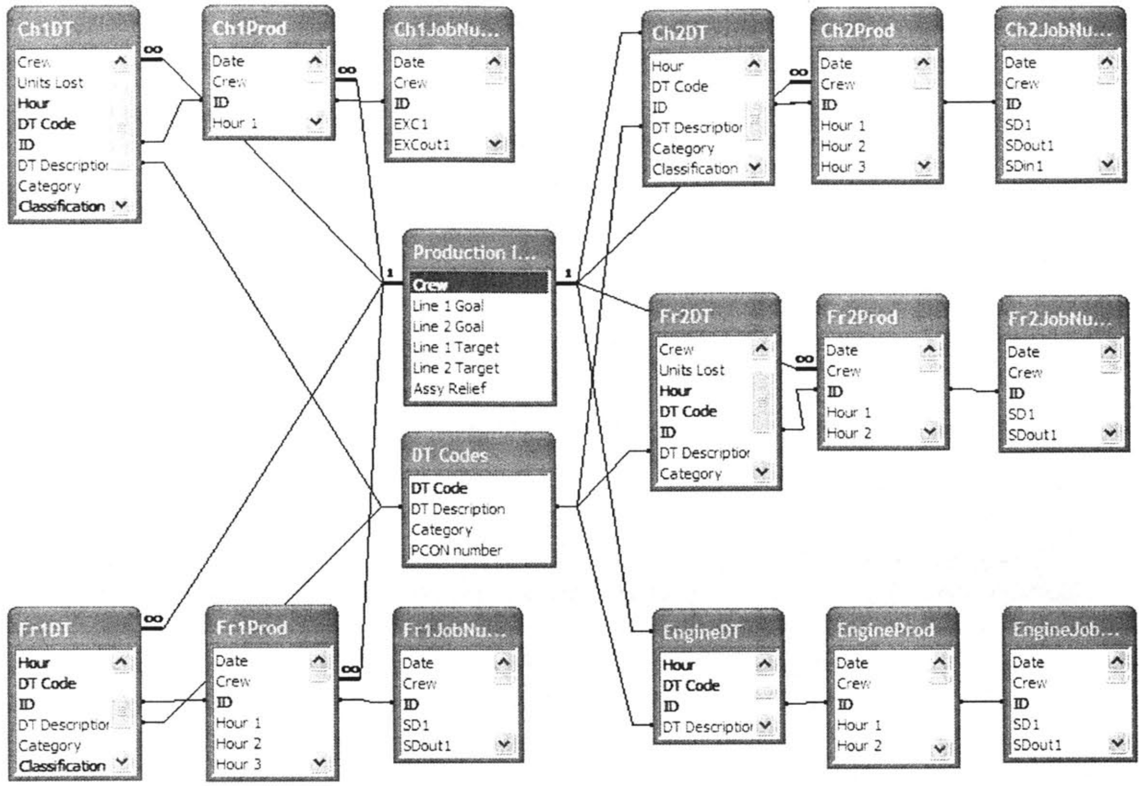


Figure 2: PICRS database ER model

From the ER model, we can see that the database consists of seventeen data tables, which names and contents are summarized in the following table.

Table Name	Table Content Description	# of Fields	Primary Key(s)
Ch1DT	This table contains information about every downtime occurred on Chassis 1 line, including fields such as Hour and Downtime Code.	11	Hour, DT Code, ID, Classification
Ch1JobNumbl	This table contains information about job numbers for both Super-duty trucks and Excursions for Chassis 1 line. One record is for one shift, which comprises of 12 hours.	81	ID
Ch1Prod	This table contains information about chassis 1 production targets and counts. Each record corresponds to one shift in production.	31	ID
Ch2DT	These tables have the same structures as the previous three tables, the difference being that these tables contain information for Chassis 2 line, and may have different number of fields because of the different types of vehicles it produce.	11	Hour, DT Code, ID, Classification
Ch2JobNumbl		42	ID
Ch2Prod		31	ID
DT Codes	This table contains all the downtime codes, their descriptions, related teams, and the PCON number where applicable.	4	DT Code

EngineDT	These tables store production and downtime information about the engine line.	11	Hour, DT Code, ID, Classification
EngineJobNumbl		42	ID
EngineProd		43	ID
Fr1DT	These tables store production and downtime information about the two frame lines.	11	Hour, DT Code, ID, Classification
Fr1JobNumbl		42	ID
Fr1Prod		31	ID
Fr2DT		11	Hour, DT Code, ID, Classification
Fr2JobNumbl		42	ID
Fr2Prod		31	ID
Production Information	This table stores information about JPH (jobs per hour) goals and targets and relief mode for each production line.		

Table 5: Data Table Descriptions

The seventeen tables can be divided into two groups. The first group includes two tables—*DT Codes*, and *Production Information*. These are the tables that store information such as downtime codes and descriptions, and jobs-per-hour targets. This information is very important to any production line and normally does not change. The second group consists of all the rest of tables. These tables store production and downtime information that grows as production carries on. Records in these tables use data fetched from *DT Codes*, and *Production Information* tables.

There are two reasons why different production lines have different set of tables to store production data: 1) this arrangement makes the database structure simpler, more elegant, and easier to understand without incurring too much structural redundancy; 2) some differences between the lines (e.g., line 1 makes both Excursion and Super Duties, while line two only makes Super Duties) make the integration of the tables troublesome. Therefore, although the principle of least redundancy may suggest using fewer tables, we found the above structure to be optimal under this particular situation.

3.1.2 Queries

The advantages of a database over a spreadsheet data reservoir are not only the more efficient means of inputting and storing data, but also the convenient ways of retrieving data and turning data into useful information. The latter tasks are achieved by well-designed queries on top of a sound database structure.

There are 36 static queries written in the database, and a dozen of dynamic queries embedded in the Visual Basic scripts. Those queries accomplish tasks as simple as "finding the corresponding possible downtimes given a team number", and as complex as "generating a Pareto report on selected number of downtimes for a given line, shift, classification, and time range combination".

The following example illustrates the design of a typical query and its output. This dynamic query takes user inputs (obtained through an input form as shown in figure 3)—namely line, date, and crew—and generates a daily production report.

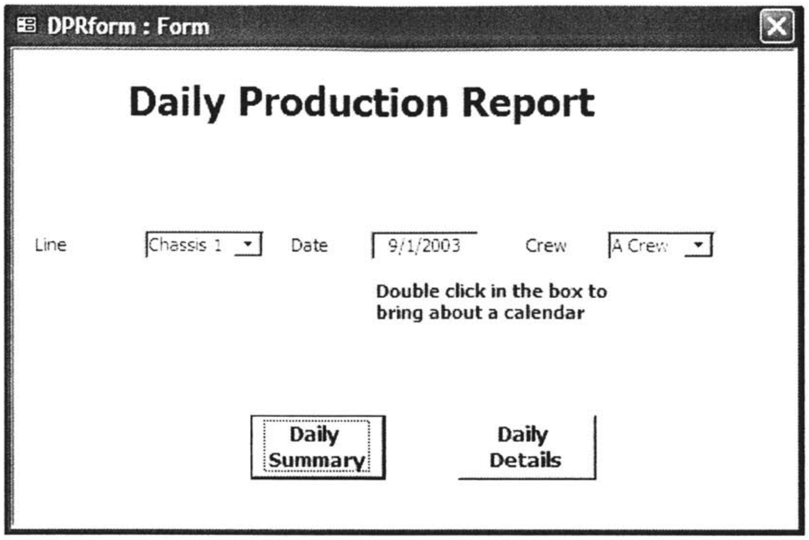


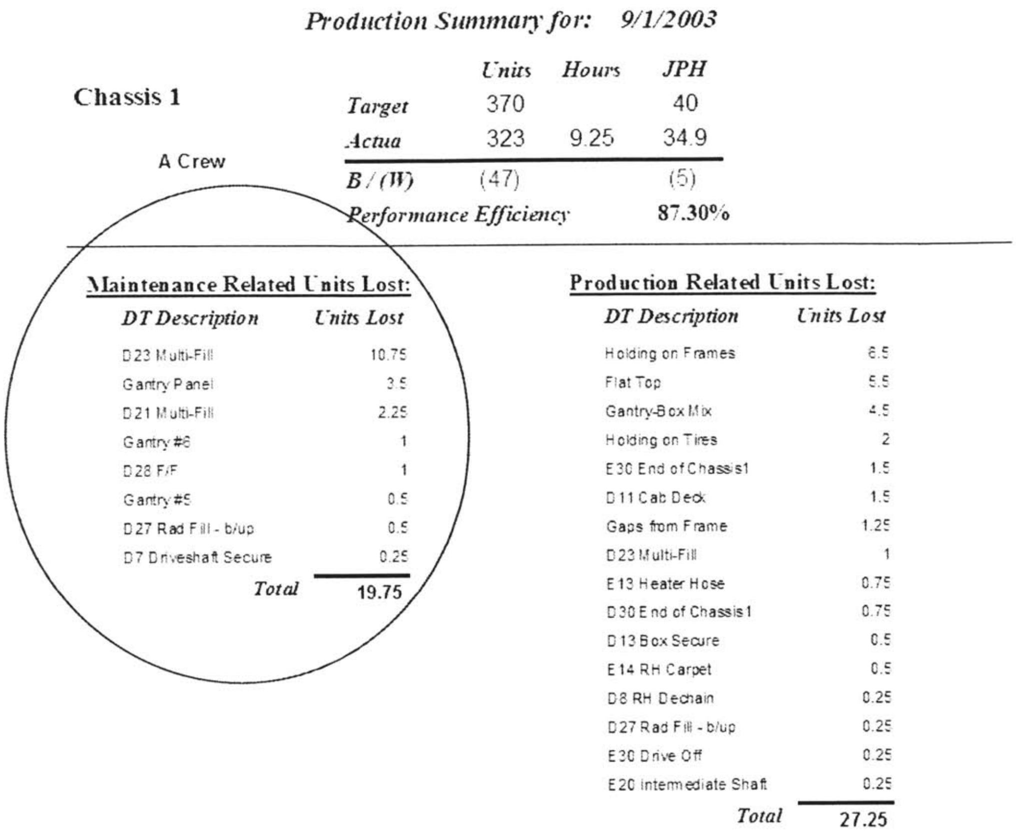
Figure 3: Input form for daily production reports generation

The daily summary for the maintenance related downtimes is generated using the following query.

SELECT date, crew, [dt description], SUM([units lost]) AS lost FROM " & dailyDetailDT *& " WHERE classification = 'equipment' GROUP BY date, crew,[dt description] ORDER BY sum([units lost]) DESC*

In this query, *dailyDetailDT* is a string variable, which is passed a value of the correct table name when the report is opened. Furthermore, upon the opening of the report, certain VB codes will dictate that the query is restricted to the records that are pertinent to the particular line, date, and crew combination inputted through the forms depicted by figure 3.

The result of the query is the circled part of the report shown in figure 4. This typical process of producing an informative and easy-to-read report involves the design of SQL query frame, the programming of VB codes that pass dynamic values to the query input, the design and programming of the data input form, and the design of the report layout.



Production Summary for: 9/1/2003

Chassis 1

A Crew

	Units	Hours	JPH
Target	370		40
Actua	323	9.25	34.9
B / (W)	(47)		(5)
Performance Efficiency			87.30%

Maintenance Related Units Lost:

DT Description	Units Lost
D23 Multi-Fill	10.75
Gantry Panel	3.5
D21 Multi-Fill	2.25
Gantry #6	1
D28 F/F	1
Gantry #5	0.5
D27 Rad Fill - b/up	0.5
D7 Driveshaft Secure	0.25
Total	19.75

Production Related Units Lost:

DT Description	Units Lost
Holding on Frames	6.5
Flat Top	5.5
Gantry-Box Mix	4.5
Holding on Tires	2
E30 End of Chassis1	1.5
D11 Cab Deck	1.5
Gaps from Frame	1.25
D23 Multi-Fill	1
E13 Heater Hose	0.75
D30 End of Chassis1	0.75
D13 Box Secure	0.5
E14 RH Carpet	0.5
D8 RH Dechain	0.25
D27 Rad Fill - b/up	0.25
E30 Drive Off	0.25
E20 Intermediate Shaft	0.25
Total	27.25

Figure 4: An example of daily summary report (numbers in this figure are fictitious for confidentiality reasons)

3.2 User Interface Design

PICRS users can be divided into three groups. The first group consists of all the final area line supervisors who use the application every hour and input all the production related data. The second group of users are the managers who use the application only to view the reports. The third group are the engineers who use the application to view reports as well as analyzing data which may require the interaction of this application with other applications such as Microsoft Excel. This said, the user interface design should serve all of the three groups of users.

3.2.1 User Interface Design for Line Supervisors

The design for the interface for line supervisors is the most demanding for two reasons. First, the line supervisors interact with the application in the most hectic working environment, and thus have very limited time to spare for using the system. Second, the line supervisors are the least computer-literate among the three groups. Moreover, since the line supervisors need to input the data every hour while managers and engineers use the system much less frequently, a crash of the system on a supervisor's workstation would have far more negative impacts than if it happened on a manager or engineer's PC.

Based on the above reasons, the main considerations for designing interfaces for line supervisors are ease of use and reliability. A description of the thoughts behind designing one of the interfaces for supervisors makes the explanation of these principles more clear.

File Edit View Insert Format Tools Data Window Help

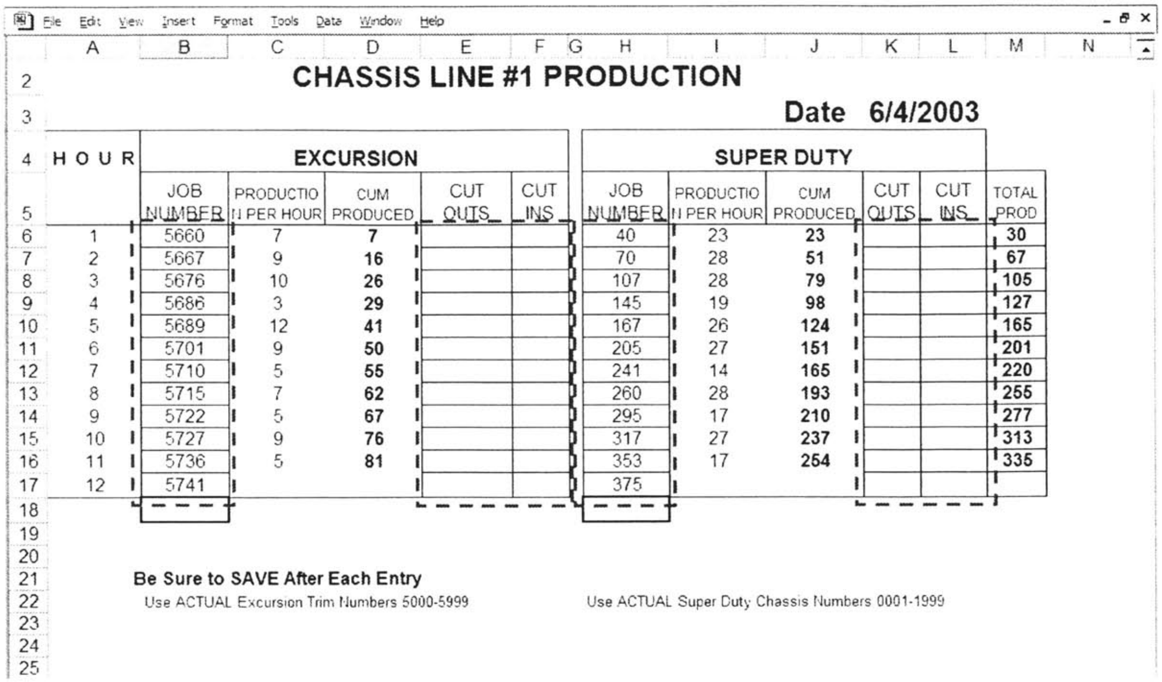
HOUR	TIME	EXCURSION UNITS PRODUCED	SUPER DUTY UNITS PRODUCED	UNITS PRODUCED	UNITS LOST	DOWNTIME DESCRIPTION	TOTAL DOWNTIME (MIN.)
1	6:00 - 7:00	7	23	30	10	8 d11 exc. Hung on crossover,2 unit gap	17.6
2	7:00 - 8:00	9	28	37	3	1 d11 gantry b/up,.5 d15 b/f,.5 d7 isolator,.5 d2 repair,.5 eol no start	5.3
3	8:00 - 9:00	10	28	38	2	.5 eol no start,.5 e16 rad.,.5 d28 f/f ot,.5 d11 gantry b/up	3.5
4	9:00 - 10:00	3	19	22	0	met production goal	0.0
5	10:00 - 11:00	12	26	38	2	.5 frame flip,.5 d25 s/r,.5 d8 unchain,.25 eol push,.25 d17 o/d	3.5
6	11:00 - 12:00	9	27	36	4	3 unit gap,.75 frame flip (vin stamp),.25 e2 repair	7.1
7	12:00 - 1:00	5	14	19	3	3 unit gap	5.3
8	1:00 - 2:00	7	28	35	5	5 unit gap	8.8
9	2:00 - 3:00	5	17	22	0	met production goal	0.0
10	3:00 - 4:00	9	27	36	4	1 unit gap.1 d11 gantry fault,.5 d22p repair,.5 e5 gun down,.5 d19 r/b,.5 d25 st mx.	7.1
11	4:00 - 5:00	5	17	18	0	met production goal	0.0
12	5:00 - 6:00	0	0				0.0
SHIFT TOTAL		81	254	331	33.0		58.2

JPH:	36.8
OEE:	66.87

DATE: 06/04/03

LINE TIME:	10.0
BREAK / min	60
MEETINGS / min	0

Figure 5: Spreadsheet for entering downtime information (numbers in this figure are fictitious for confidentiality reasons)



CHASSIS LINE #1 PRODUCTION

Date 6/4/2003

HOUR	EXCURSION JOB NUMBER	EXCURSION PRODUCTION PER HOUR	EXCURSION CUM PRODUCED	EXCURSION CUT OUTS	EXCURSION CUT INS	SUPER DUTY JOB NUMBER	SUPER DUTY PRODUCTION PER HOUR	SUPER DUTY CUM PRODUCED	SUPER DUTY CUT OUTS	SUPER DUTY CUT INS	TOTAL PROD
1	5660	7	**7**			40	23	**23**			**30**
2	5667	9	**16**			70	28	**51**			**67**
3	5676	10	**26**			107	28	**79**			**105**
4	5686	3	**29**			145	19	**98**			**127**
5	5689	12	**41**			167	26	**124**			**165**
6	5701	9	**50**			205	27	**151**			**201**
7	5710	5	**55**			241	14	**165**			**220**
8	5715	7	**62**			260	28	**193**			**255**
9	5722	5	**67**			295	17	**210**			**277**
10	5727	9	**76**			317	27	**237**			**313**
11	5736	5	**81**			353	17	**254**			**335**
12	5741					375					

Be Sure to SAVE After Each Entry

Use ACTUAL Excursion Trim Numbers 5000-5999

Use ACTUAL Super Duty Chassis Numbers 0001-1999

Figure 6: Spreadsheet for entering job numbers (numbers in this figure are fictitious for confidentiality reasons)

Figures 5 and 6 are screenshots for sample spreadsheets used for production data input before the PICRS was implemented. The areas encircled in the dashed blocks are the cells that require input from the line supervisors. The encircled area in figure 5 records downtime information including *units lost*, and *downtime description*. The encircled areas in figure 6 record *job numbers* and *cut-in/cut-out* units (additions to or removals from the original production plan). Formulas are written in certain cells of the spreadsheet template to calculate each hour's production counts for both super-duty trucks and Excursion SUVs produced. Aside from the inconvenience that one separate Excel file has to be created for each day-shift combination, there are several problems with using the above spreadsheet to input production information:

- There is no mechanism to caution against, or prevent input errors.
- The downtime descriptions are time-consuming to enter. As a result, the supervisors sometimes "simplify" this step by entering very short descriptions or even skipping recording certain downtimes.
- The descriptions of the same downtime vary by each supervisor. This makes it hard to consolidate all the information afterwards.

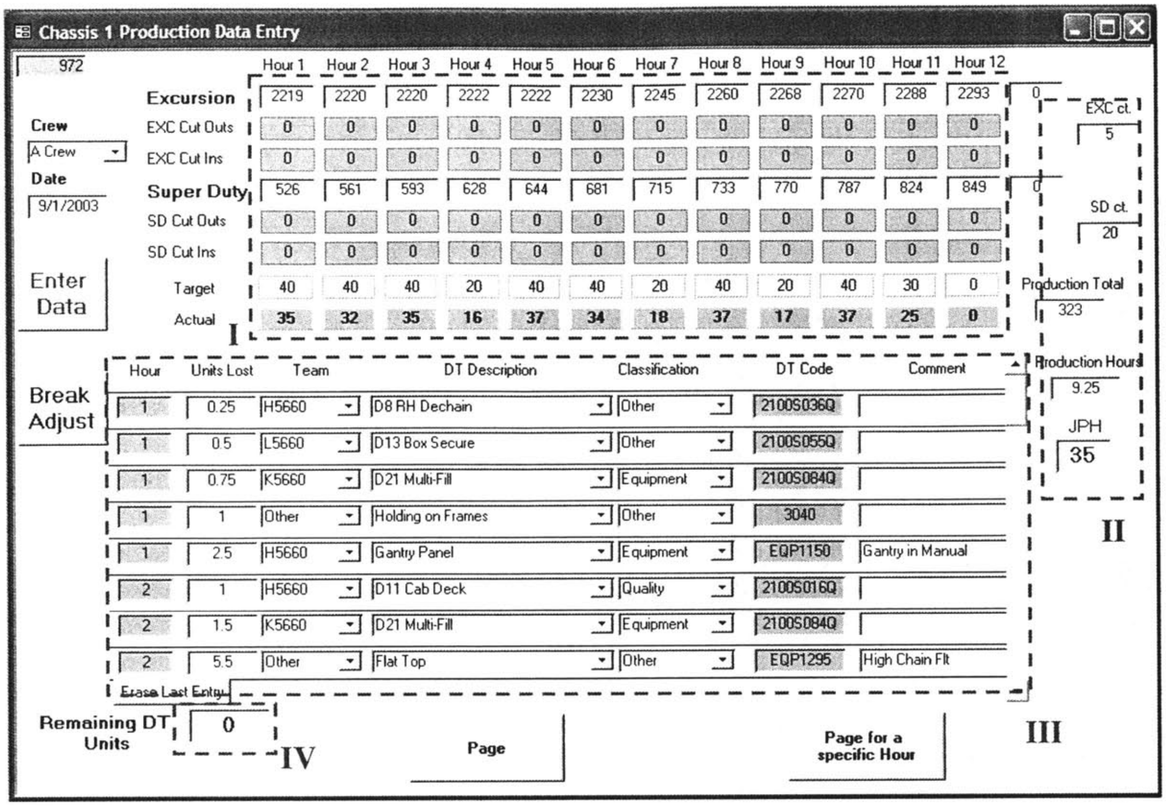


Figure 7: Sample PICRS input screen (numbers in this figure are fictitious for confidentiality reasons)

The PICRS input screen (Figure 7) can solve the above mentioned problems associated with using spreadsheets, and can bring additional advantages as well. The supervisor uses this input screen for the whole duration of the shift. He first needs to select the data/crew combination located at the upper left corner of the form. This input activates all the fields on this screen and prompts the system to fill in all the default values and retrieve last shift's final job numbers, respectively for Excursions and Super-duty trucks, to appear as the current shift's first two job numbers. In this screen, only the white text boxes and dropdown lists in area I and III need input or selection from the user. The *cut-ins* and *cut-outs* fields are pre-filled with zeros, but can be modified when occasionally some planned schedule changes. The actual production numbers, as indicated in the bottom line of area I, are automatically calculated from the job number input in the white text boxes. Text boxes in area II provide the user with several pieces of useful information, such as numbers of Excursions produced in the current hour, cumulated total production, and cumulated JPH, etc. This information is also calculated from using the job numbers entered by the user with the help of arithmetic formulae imbedded in those textboxes. In area III, the user can, instead of typing in the description, select the downtime description from a dropdown list filtered by the team he has chosen. By doing this, the supervisor saves time as well as ensures the consistency of downtime description input by all the users. Moreover, with the *Classification* field and an optional *Comment* field, the downtime information collected by PICRS is more detailed. The yellow text box in area III tells the users how many more lost units he needs to account for, thus providing a means to caution against input error.

The Classification field classifies each downtime entry as one of the following six types described in table 6.

Classification Value	Meaning
Equipment	The downtime is equipment related and maintenance personnel need to respond to this type of downtime
Less Stock	The downtime is caused inefficient material handling
Man Power	The downtime is caused by the absence of operators or maintenance personnel
Operator	The downtime is due to the operator's faulty operation
Quality	The downtime is due to a faulty part or unqualified material
Other	All other reasons that don't belong to the above categories

Table 6: Explanation of the Classification field

This PICRS input form also has other handy functions such as *break adjust*, and *page through email*, which were not available with the spreadsheets. The *break adjust* function provides a handy way to adjust hourly production targets in case of an unplanned break, such as for an emergency meeting. Figure 8 shows a snapshot of the Break Adjust form. The *page through email* function enables the supervisor to send text pages about the current hour's production number and downtimes to plant management by simply clicking one of the two buttons at the bottom of figure 7.

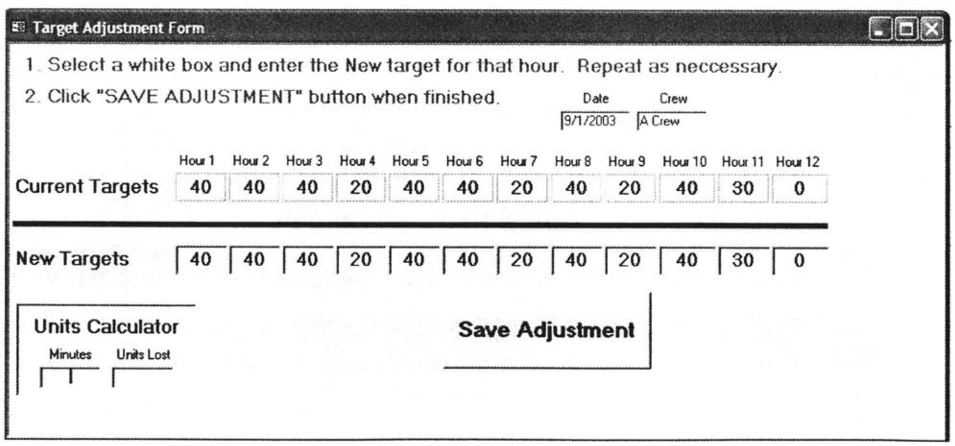


Figure 8: The Break Adjust form

Besides all the benefits PICRS provides for data entry, the more powerful functions come from the capability of backend relational database. All the data input goes into the database, which makes the storage and manipulation of the data a lot easier.

3.2.2 User Interface Design for Managers and Engineers

Managers use PICRS primarily for viewing and printing reports. The capability to manipulate data and generate various useful reports is the main consideration in designing the user interfaces for managers.

Figure 9 shows a sample interface for crew and area managers to use to generate downtime pareto reports. Using this form, the manager can choose any line-crew-classification combination, any time range, and a number between 1 to 10 or "all" to designate how many downtime reasons to be shown. She can also choose either list or chart format for the report.

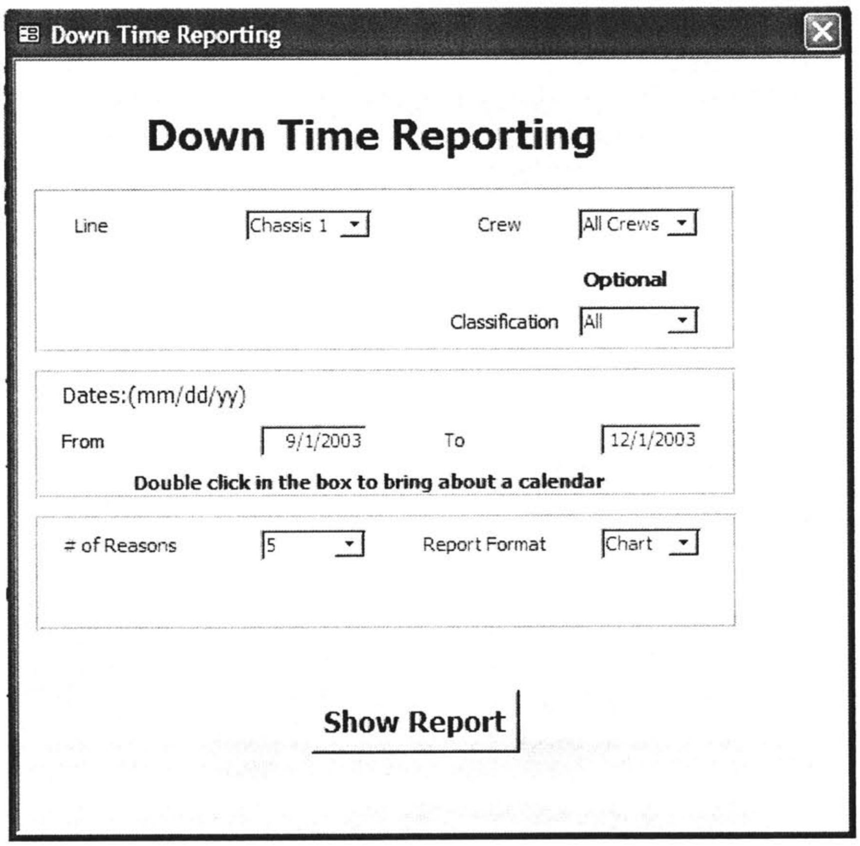


Figure 9: Sample user interface for managers

To facilitated the date input process and make the input consistent, we also built in a programmed calendar as shown in figure 10. The programmed calendar is a small piece of free software that is available from the Internet. By using available free software, we saved some development time.

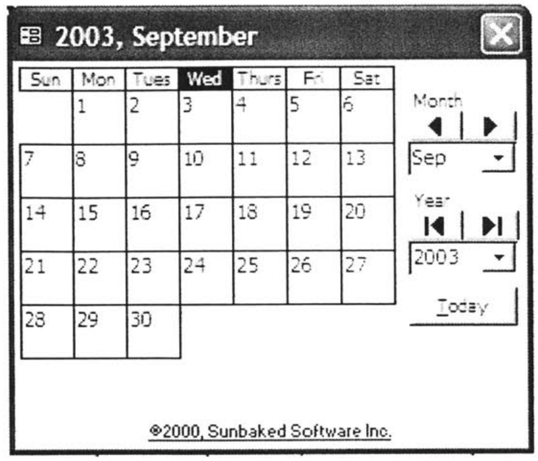


Figure 10: Calendar to facilitate the date selection

Figure 11 shows what the report looks like based on the input in figure 9. This is only one example of the many reports that managers can generate with easy-to-use interfaces like the one in figure 9. Feedback has shown that managers have found these reports to be very handy and useful (see section 5.1 of this thesis for some quotes of the user feedback). This functionality of PICRS is something that the old spreadsheet application does not have at all.

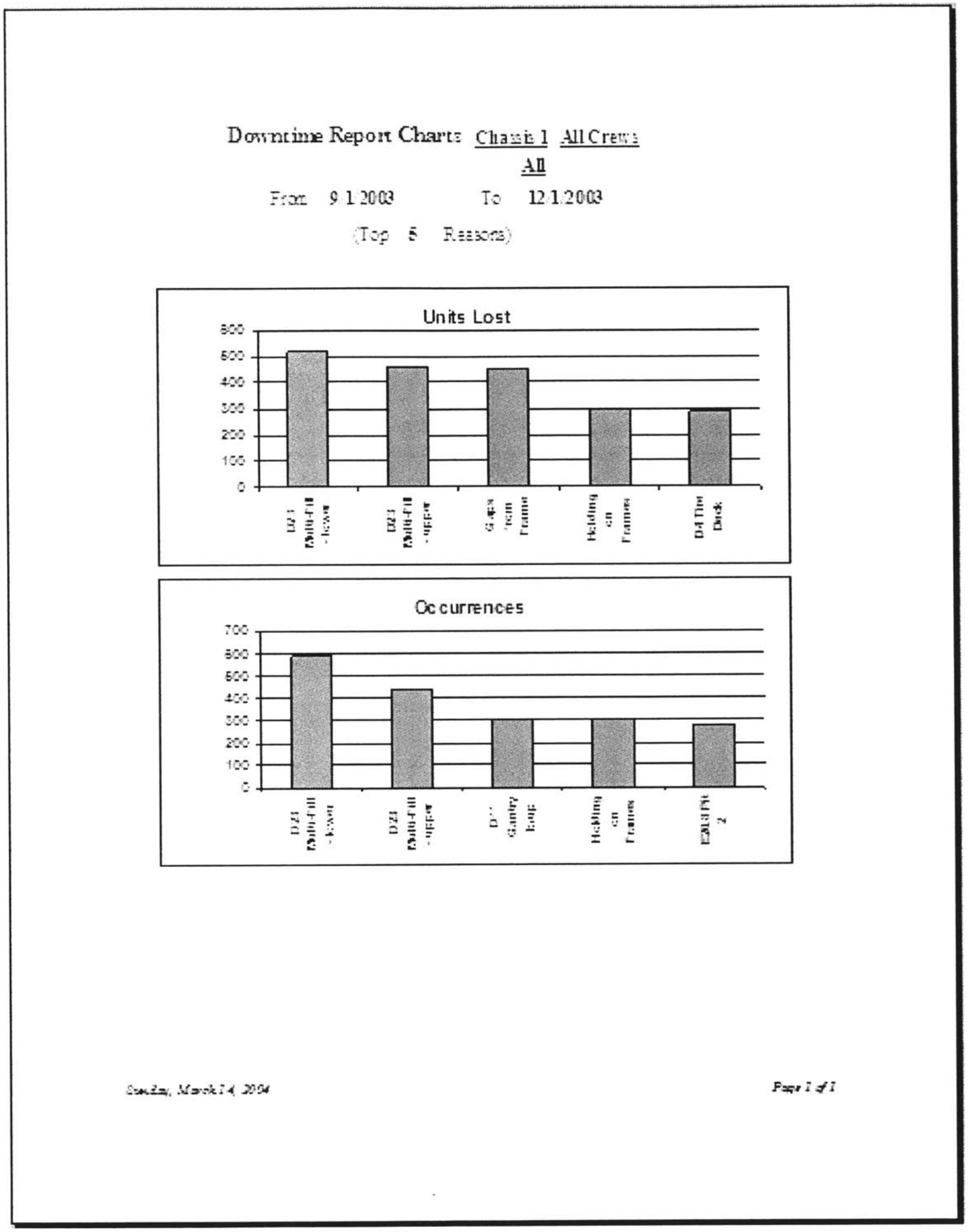
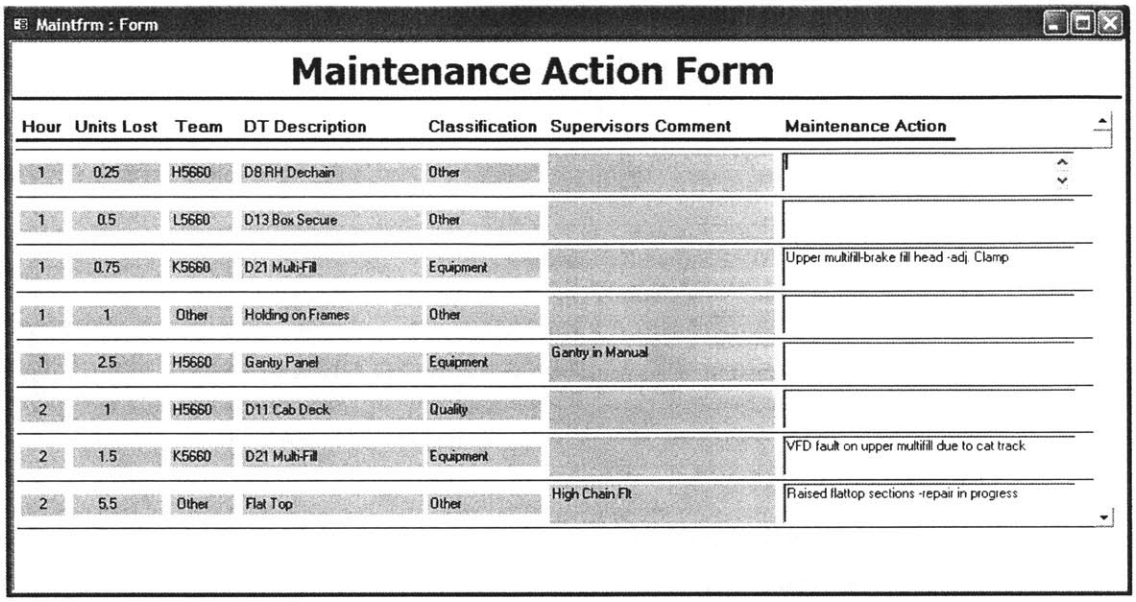


Figure 11: Sample report for downtime pareto charts (numbers in this figure are fictitious for confidentiality reasons)

Engineers, including industrial engineers and maintenance engineers, interact with PICRS more analytically. Maintenance engineers need to enter maintenance comments to each downtime that is equipment related. We have created fields in the data tables and separate input forms to enable maintenance engineers to do so. Figure 12 shows the interface where maintenance can enter comments on line downtimes in a real-time fashion. Managers can also view these comments in the reports they generate. This functionality greatly enhanced the communication between production and maintenance people and is warmly received.



Maintfrm : Form

Maintenance Action Form

Hour	Units Lost	Team	DT Description	Classification	Supervisors Comment	Maintenance Action
1	0.25	H5660	D8 RH Dechain	Other		
1	0.5	L5660	D13 Box Secure	Other		
1	0.75	K5660	D21 Multi-Fill	Equipment		Upper multifill-brake fill head -adj Clamp
1	1	Other	Holding on Frames	Other		
1	2.5	H5660	Gantry Panel	Equipment	Gantry in Manual	
2	1	H5660	D11 Cab Deck	Quality		
2	1.5	K5660	D21 Multi-Fill	Equipment		VFD fault on upper multifill due to cat track
2	5.5	Other	Flat Top	Other	High Chain Flt	Raised flattop sections -repair in progress

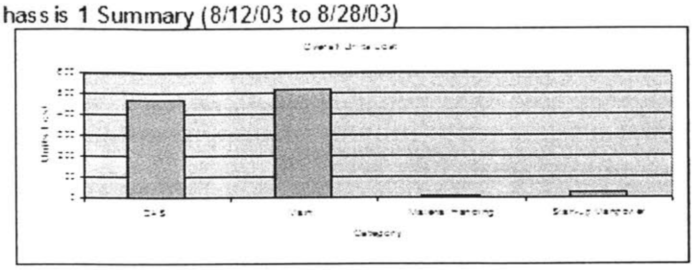
Figure 12: PICRS interface for maintenance engineers (numbers in this figure are fictitious for confidentiality reasons)

Industrial engineers use production information to produce various reports on matters like OEE (Overall equipment effectiveness) and periodic production analysis. PICRS is an effort to automate some of the information processes at KTP. But due to the time and budget constraints of the project, many information processes have to remain the old, Excel-based way. However, we did create some interfaces and wrote VB codes for industrial engineers to automatically populate certain Excel-based reports using data stored in PICRS.

Figure 13 is part of the OEE report automatically generated by PICRS upon clicking a few buttons by the industrial engineers. Before PICRS, it took one industrial engineer at least half a work day to come up with the same report.

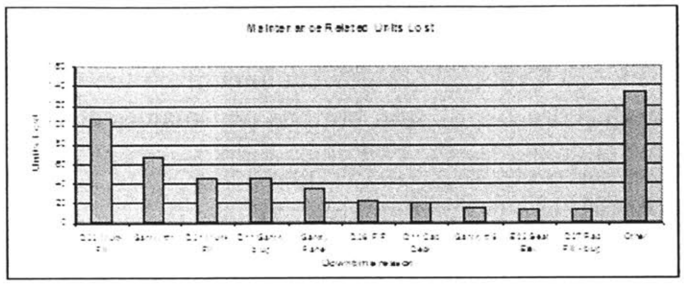
Chassis 1 Summary (8/12/03 to 8/28/03)

Overall	1023.5
QAS	467.25
Maint.	523
Material Handling	7.75
Start-up Manpower	25.5



Maintenance

Total	523
D23 Multi-Fill	106.5
Gantry #1	67.5
D21 Multi-Fill	46.25
D11 Gantry b/uc	46.25
Gantry Panel	36.75
D28 F/F	22
D11 Cab Deck	21
Gantry #6	16
E23 Seat Elev	14
D27 Rad Fill - b/uc	13.75
Other	134



Quality Stops

Total	467.25
Gantry-Box Mix	55
CMWG	36
EOL No Starts	35.5
Holding on Engines	34
Gaps from Frame	29
Engine Deliv	22.5
E20.8 Pit 2	19.5
D4 Tire Deck	18
Holding on Cabs	17
Holding on Frames	14.25
Other	255.25

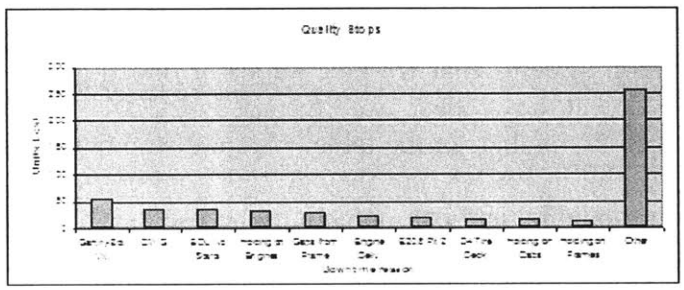


Figure 13: Partial sample Excel report automatically generated by PICRS (numbers in this figure are fictitious for confidentiality reasons)

3.3 Downtime Codes Structure

One of the major drawbacks of the old Excel-based system is that there is no consistent way to describe a downtime. Individual supervisors use somewhat different descriptions for the same downtime reason. For example, for the same downtime reason such as *Radiator Install Failure*, one supervisor might write "Radiator install down, 2 U"—meaning the time to produce two trucks (if at full speed) is lost due to failure at radiator install station—while another supervisor might just write "2 Rad Inst.". Differences in descriptions, such as in this case, make automatic manipulation of downtime data, such as grouping and sorting, impossible. In designing the PICRS system, a major task is to develop a list of downtime codes that are well structured, informative, comprehensive, and easy to use.

From the beginning, we thought that PICRS could not only automate the reporting process, but also be a good opportunity to provide information with the level of detail that has never been achieved before.

Any stop at the continuous assembly line can be attributed to one of two types of reasons: production related and equipment related. We define production related downtimes as those that are triggered by the push of one of the stop buttons on the lines. The reasons for an operator or a supervisor pushing a stop button are numerous, and include the observation of imminent equipment failures. However the definition is very clear that any stop that is triggered by a stop button is categorized as a production related stop. On the other hand, an equipment related stop is one that is triggered not by a stop button but by the failure of a piece of equipment.

After categorizing two types of stops, we went out to map each and every stop button, and each piece of major equipment on the two chassis lines. This rather tedious effort produced a list of stop buttons and major equipment that became the basis of our downtime code list for the two chassis lines.

We created downtime codes, which are basically short descriptions of downtime reasons, pertinent to each stop button as well as each piece of major equipment. In the DT Codes data table, we recorded the stop button number for each production related downtime code, in the hope that some time in the future we can link PICRS to other information systems that are electronically linked to those stop buttons. For equipment related downtime codes, we recorded PCON number, which is the equipment identification number, in the table. Like the stop button IDs, the PCON numbers are not readily useful in PICRS, but they do link equipment IDs to equipment related downtime codes in the table, and can potentially be useful if PICRS is linked to other information systems that use PCON in the future.

Locations, i.e. factory pillar numbers, are also provided in the production related downtime codes. For example, now we have, in PICRS, "D16 Radiator Install" and "E16 Radiator Install" for two production related downtime codes (two stop buttons, D16 and E16 are pillar numbers). The supervisors used to only write a short description such as "Rad Inst" for any of the two reasons. Now the information is more clear and detailed.

Occasionally, there is a downtime due to unusual reasons, such as power failure. We include in the downtime reason list a number of "common" unusual downtime codes which are not categorized as either production related or equipment related. However, it is impossible to include all the possible unusual stop reasons. In this case, we have a reason code named "other", and the supervisor has to write a description in the Comment field to specify.

Besides the downtime description itself, we also create several other fields to supplement a downtime entry. These fields are described in the following table.

Field	Description
Team	The ID number of the team where the downtime happens. It is selected first by the user as a filter to the numerous downtime codes. The average number of downtime codes filtered by a team number is about 10.
Classification	This field classifies each downtime code into one of the six categories: *Equipment, Less Stock, Man Power, Operator, Quality*, and *Other.*
Comments	This is an optional field. The supervisor can input any additional information he thinks is important to this downtime.

Table 7: The descriptions of additional fields in a downtime record

Table 8 shows two sample downtime entries with both the old Excel-based system and PICRS about the same downtime reason. We can see that the downtime entry with PICRS contains much more information. Since it only takes a few clicks to input the downtime information with PICRS, it actually takes less time to record more information with PICRS after the user become adept in using the new system (the users can also use keyboards instead of clicking the mouse).

DT reason	Lost two units (the time to produce two trucks at full speed) because pillar F22 left-hand seat secure stop button was pushed					
Old system	F22 Seat Secure 2.0					
PICRS	DT code	Team	DT Description	Classification	Units Lost	Comment
	200S153Q	D5670	F22 lhs Seat Secure	Quality	2.0	

Table 8: Comparison of the downtime description with the old system and that with PICRS

There are limitations with PICRS downtime entry mechanism. A few supervisors did not find it very easy to adapt to the new system. They complained that it took them slightly more time to input downtime information than they used to with the Excel-based system. This is, however, not the case with the majority of the users. Some supervisors entered more detailed information then others with the old system. Using PICRS, these supervisors chose to use the *Comment* field more often than others. To them, PICRS is not time-saving.

All of above descriptions are about the design of downtime codes for the two chassis lines. While information about stop buttons and equipment is available for the chassis line, this is not the case with other production lines in the final area. For the two frame lines and the engine line, we composed the downtime codes using the assembling jobs and their locations. These downtime codes are more "primitive" than those for the chassis lines. However, with the old Excel-based system, frame line downtime information was recorded with much less effort, and engine line downtimes were not even recorded. So, PICRS is also an apparent improvement over the old way, if there was any, that frame and engine line downtime was recorded.

3.4 Summary of System

In summary, PICRS has 17 data tables, 35 static queries, 40 programmed forms (including sub-forms), 26 report templates (including sub-report templates), and a total of 7306 lines of Visual Basic for Applications codes. There are 615 downtime reason codes embedded with the system.

Chapter Four: System Implementation

PICRS' evolutionary process of system development and its incremental implementation proceeded in parallel. While the time per week spent in debugging, revising, and upgrading the system was decreasing drastically during the process, the implementation accelerated greatly. For instance, it took about two weeks to implement PICRS onto the chassis line, while it only took three days to implement it onto the engine line. The concurrent development-implementation process proved to be conducive to learning for both the developers and the users.

During the implementation process, the developers' interactions with the users also produced observations on matters such as company culture and effective communication. This chapter is an account for the implementation process and some of the observations.

4.1 Implementation Considerations

Before the first prototype of PICRS was implemented onto the two chassis lines, there were several concerns. First, we did not know how well PICRS can handle when multiple users try to write information to and retrieve information from the system, although an unqualified experiment did not indicate any problem. Second, we did not know how quickly and easily the supervisors would adapt to the new system. Third, we did not know how stable and robust the system could prove to be. To put it another way, we did not know how often the line supervisors, most of whom with minimal computer skills, would screw up the system. Fourth, we were not sure if the distributed arrangement of the system would have any compatibility issues with the software and hardware of the dozens of computers that PICRS would reside in.

Our first concern was gradually eliminated as we installed the front-end of PICRS on multiple computers. As mentioned, we experimented with simultaneous operations of PICRS from the two PCs of the developers and found no problem. On the first day of the implementation, we installed PICRS front ends on four workstations on the two chassis lines. Two of the four computers are primary ones for PICRS in that the supervisors normally only use these two for data input. On the first day, two supervisors interacted with PICRS at the same time from two terminals, and the system worked properly. As we expanded the system onto more computers, more users were interfacing with the database simultaneously using various input forms. No problem concerning simultaneous data and form manipulation ever occurred.

The supervisor's adaptation to the new system varied by each individual. Overall, the supervisors learned how to use PICRS much faster than we expected. Young supervisors with college degrees learned the system especially fast. They were quickly using hotkeys instead of the mouse. They were also the main sources of suggestions on how to improve the system. Even older and less computer-literate supervisors adapted to the new system pretty quickly. Although they now may need to spend slightly more time inputting data than they did in the old way, they acknowledge that PICRS is overall an improvement over the old Excel-based system.

In the first two weeks of the implementation, problems with proper using of the system happened more frequently than later on. On average, we were called once or twice a day to go to the lines to solve problems with PICRS. Some of the problems were with the system itself, and solutions were debugging and upgrading of the system. Other problems were due to the improper operations from the users. Before implementation, we drafted a user's manual and distributed it to all the users. However, we did not have the authority to force the supervisors to read it, and we suspected that very few of them did. But thanks to the simplicity and ease-of-use of the system, the issue of improper use did not turn out to be a big one. The problem was solved on an individual basis in several weeks.

The concern about compatibility between PICRS and different operating system versions turned out to be a justifiable one. Actually, in the first month after the start of the implementation, no compatibility issues appeared. After we had implemented PICRS onto both chassis and frame lines, the system department of KTP replaced some workstations on the lines as well as some PCs for the engineers with more powerful computers with newer versions of Windows operating systems. A problem with the date format arose due to the different versions of the Windows operating system. We solved the problem by installing an extra Windows Access Object Library for all the front-end forms to reference to.

4.2 The Implementation Process

As mentioned before, we adopted the evolutionary prototyping method for PICRS system development, and an incremental expansion method for the implementation of the system. Figure 14 shows the timeline of the whole internship project. The bold texts indicate milestones of the project, and other texts describe the time intervals spent on specific tasks. The implementation process is indicated by the underlined text and a bracket to show the duration.

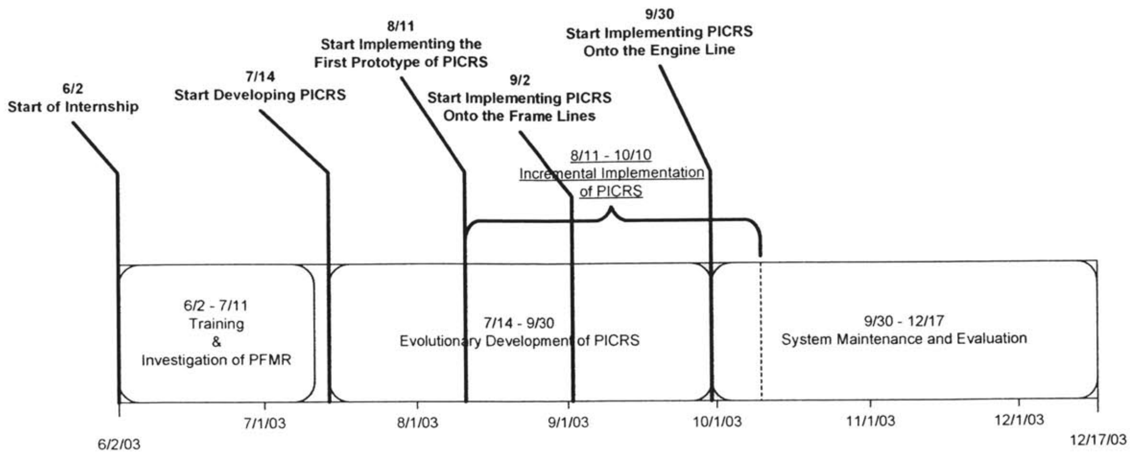


Figure 14: Timeline of internship project

The implementation started four weeks after the start of PICRS development. This incremental approach for both system development and implementation entails some overlap of the development and implementation processes. The incremental approach saved time and ensured that the final system be the one that met all the user requirements.

The day before the implementation of the first prototype of PICRS onto the chassis lines, we conducted two 20-minute training sessions for the chassis line A crew supervisors. The A crew for both chassis lines started to use the system on August 11th. We let A crew alone to use the new system until August 17th, when we started to train C and B crews to switch to PICRS.

On the first day of implementation, we used the "double-entry" method to ensure smooth transition—while the supervisors used PICRS to record the information, we developers input the same data into the old Excel templates. After the first several days of implementation, we programmed a VB scripts to automatically populate the spreadsheets using PICRS data entered by the supervisors. In this transition period, we used both old and new systems. We abandoned using the old system about one week into the implementation when we had gained confidence of the proper functioning of PICRS.

Encouraged by our experience with the implementation on chassis lines, we speed up the PICRS implementation on the frame lines in the following ways. First, we did not hold training sessions; instead, we trained the supervisors as we implemented the system. Second, we abandoned the old Excel-based system at the outset. Third, only one of us (in fact the author himself) conducted the implementation. I stayed for two consecutive shifts to train both A and B crew supervisors and came back to continue the training when C crew supervisors came in late that week. We were able to greatly accelerate the implementation on the frame lines not only because we had climbed up the learning curve, but also because the downtime codes for frame lines were significantly simpler than those for the chassis lines.

The Engine line PICRS was actually an add-on to the project. Before PICRS, the engine line did not have any system to record and keep production related data as chassis and frame lines did. After seeing benefits PICRS brought to chassis and frame lines, the crew manager of engine line (who is also in charge of the frame lines) requested that the system also cover engine line. We did as requested, and the author installed the program and trained the engine line supervisors in three days.

4.3 Change Management and Leadership Lessons

The LFM internship projects are not only activities in which LFM students practice what they have learned at school and create value for the MIT-industry partnership, but also opportunities in which the students can develop leadership skills by creating positive changes in the host companies.

Leadership is an integral part of the LFM program and is also one of the focuses of the 6.5-month internship project. My internship project involved, in the order of natural progress, analyzing KTP's need of a more efficient production data reporting system, evaluating alternative solutions, developing PICRS, and implementing the new system. Among those tasks, the implementation of PICRS required most interactions with people. The implementation process was where I created real changes on the KTP plant floor, and also where I met most leadership challenges.

4.3.1 The Sloan Leadership Model

A research group led by Professor Deborah Ancona at MIT Sloan School of Management argues that leadership is a process to create change. They further postulate that leadership is about making things happen, contingent on a context[10]. This group of researchers also developed a leadership model that is founded on the three Cs: *catalyzing action*, *contingent on context*, and *change signature*. Figure 15 is a pictorial representation of the model which also depicts a frame work to understand leadership. The framework revolves the identification of four key leadership capabilities: *Sense-making*, *Relating*, *Visioning*, and *Inventing*. The framework also includes a notion of Change Signature—ones own way of create changes.

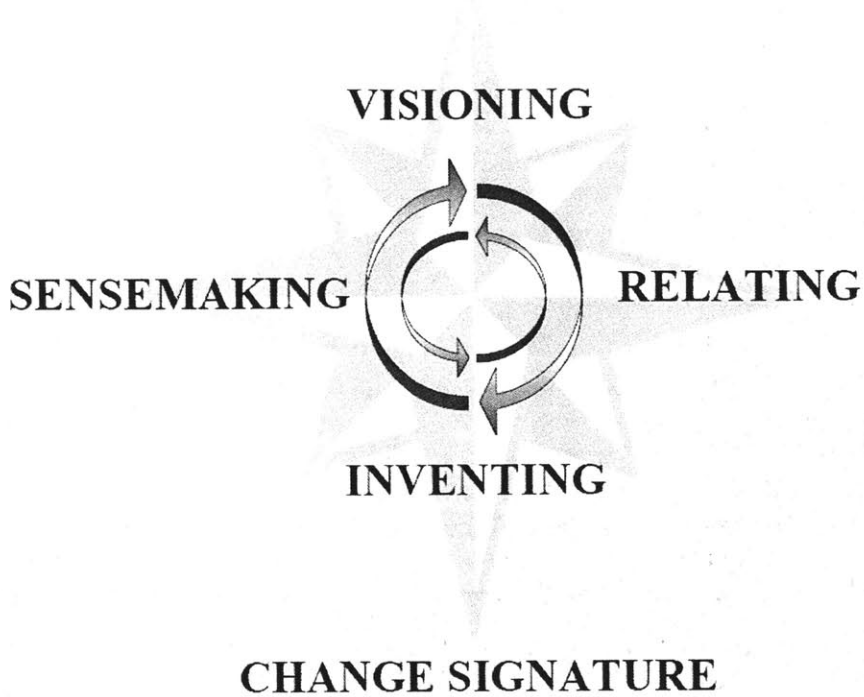


Figure 15: The Sloan leadership model

The meanings of the four leadership capabilities can be summarized as follows[11]:

- **Sense-making:** Triangulates a wide variety of data about organization and stakeholders, actively surfaces others views, and creates a new map of what is happening in the group or organization.
- **Relating:** Listens to others, encourages expression of diverse viewpoints, advocates own point of view to others, values and develops others, and builds networks of collaborative relationships with others.
- **Visioning:** Creates compelling vision for others, builds follower support, and shows the way through expressing passion and modeling behaviors that support the future vision.
- **Inventing:** Invents new modes of work, encourages experimentation and risk, coordinates change processes, monitors results, and creates an atmosphere that helps others to produce.

I have found this framework particularly helpful for me to consciously practice and develop my leadership skills while implementing changes during my internship project.

During the first five weeks of my internship (see figure 14 for the internship timeline), I spent most of my time *sense-making* and *relating*. It was crucial for me to understand the situation in order to decide what next step I needed to take. Firstly, I needed to understand how badly KTP was in need of timely and accurate production information and how inefficient the old system was. Secondly, I had to know what alternative solutions were available and how well they could meet the needs. A clear alternative was the corporate-developed web-based Plant Floor Metrics Reporting system (PFMR). We spent weeks investigating the system only to find that it had too few functions and was too cumbersome for the line supervisors to use. In testing PFMR, we also figured out the map of the stakeholders of this failed system (who were the advocators, who were the opponents, and what were the reasons for advocating and oppositions?). It was a good practice of sense-making. Thirdly, after ruling out the possibility of using PFMR, we had to gather the user requirements for the development of a new system. At the same time, we had to work with management to gain their support.

Relating played an important role in gathering user requirements and securing management support for the development of PICRS. We listened to each group of potential users and advocated to them the benefits PICRS could bring. In doing so, we tried to encourage the potential users to provide constructive suggestions. When dealing with the management, we articulated the infeasibility of PFMR and provided a *vision* of what PICRS could do for KTP. After we made sure that our messages went across clearly, we tried to establish cooperative relationships that would later help with a smooth implementation of PICRS.

The system development process exemplified our adherence to the fourth capabilities of the leadership framework—*inventing*. The idea of developing a system by ourselves (two interns) and implementing it with only a few months was itself bold and inventive in that context. After all, a corporate-developed system (see section 2.3.2 for brief description of this system called PFMR) that had cost huge amount of time, labor, and money had just recently failed to be effective. In choosing software solutions for the system (see section 3.2 for more information on software consideration), we invented by adopting the non-conventional approach—instead of using the popular web-based interface, we opted to use the faster and more flexible client-server structure. This choice turned out to be a sensible one, as we later added functions to the input forms. Throughout the development process, we were unconstrained by designing and programming norms. Our paramount guiding principle was to develop a system that could provide the functions KTP needed, was easy to use, and cost little. To conform to this guideline, we tried to be flexible and inventive in many of the development tasks. Utilization of available free software to expedite the development process is good example of our being flexible in serving the main goal (see section 3.2.2 for more information about this example).

4.3.2 The PICRS Stakeholder Analysis

A change process usually involves multiple stakeholders. As for PICRS development and implementation the stakeholders include the sponsors, the developers, and the end users. Each group is further composed of various individuals. For example, the user group includes the line supervisors, industrial engineers, and area managers. In total there are over 50 individuals involved in the PICRS project.

Analyzing stakeholders' commitments to the project can often shed light on a more effective approach to accomplish the project. Beckhard and Harris[12], in their book

Organizational Transitions, describes a method called *commitment charting* used to form a diagnosis and action strategy to get necessary commitment from target individuals or groups identified as critical mass for change implementation.

Table 9 is the stakeholder commitment chart for the PICRS development and implementation. We rate the level of commitment into five categories, ranging from "against change" to "make it happen". The table provides the information about the initial stances (the X's) of all the stakeholders and the levels of their commitments (the O's) as required for a successful project. Please note that it is sufficient to get minimum commitment deemed necessary from each individual or group. It is sometimes unrealistic or even wasteful to get everybody's commitment to the level of "make the change happen". The arrows in the chart indicate the needed transitions of stakeholders' level of commitment.

Key Stakeholders	**Against Change**	**No Commitment**	**Let It Happen**	**Help It Happen**	**Make It Happen**
Plant Manager			XO		
Manufacturing Engineering Manager				XO	
Production Engineering Manager				XO	
IT Manager	X →	→	O		
Area Manager		X →	O		
Crew Manager 1		X →	→	O	
Crew Manager 2		X →	→	O	
Line Supervisors		X →	→	→	O
Industrial Engineers			XO		
Developers and Implementers					XO

Table 9: PICRS stakeholder commitment chart

The chart clearly indicates that the successful implementation of PICRS needs higher level of commitment from the IT manager, the area manager, the crew managers, and the line supervisors.

Figure 16 is PICRS stakeholders' map which shows how individual stakeholders and stakeholder groups interact with each other to create the critical mass for the implementation of PICRS.

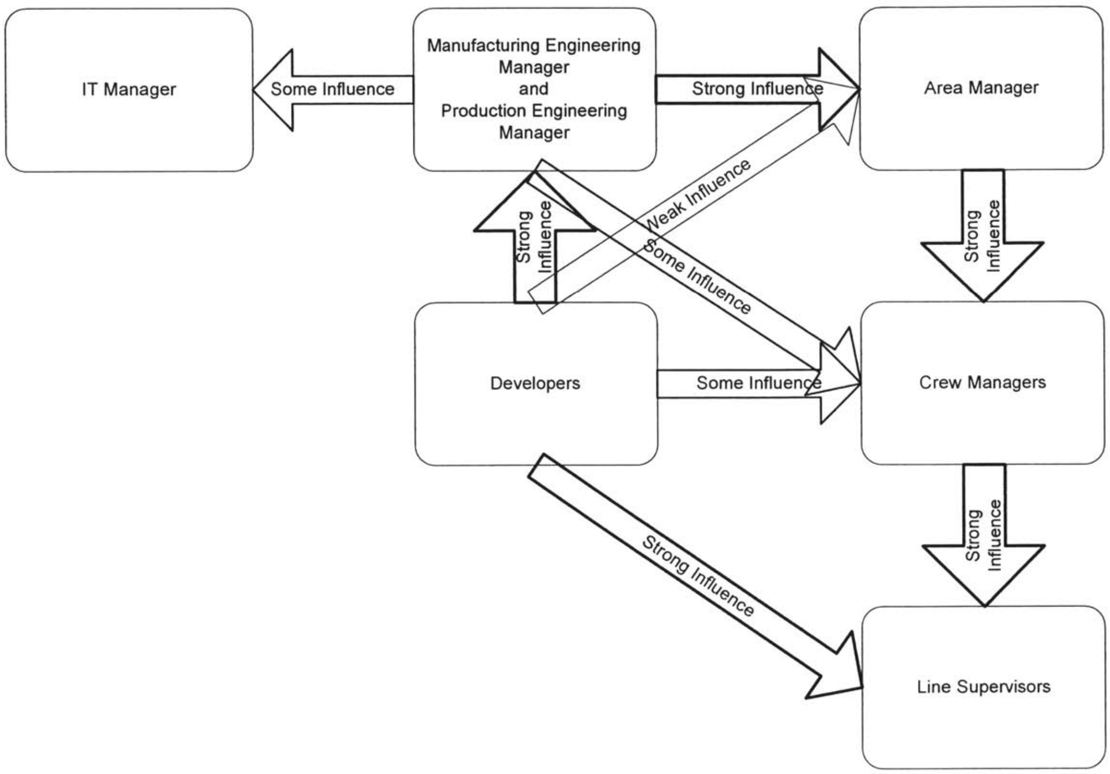


Figure 16: PICRS Stakeholders' Map

In the map, the blocks are individual stakeholders or stakeholder groups, and the arrows are the directions of influence. It is quite clear that the developers of the system, the manufacturing engineering manager, and the production engineering manager are the initial main advocators of PICRS. The other four stakeholders (groups) are the ones that need to be moved up the level of commitment.

The manufacturing engineering manager and production engineering manager are the sponsor and supervisor of the internship project. They got involved from the beginning and, even before the implementation, had a good understanding of how PICRS could benefit KTP. Their commitments were partly from their understanding of the usefulness of the project and partly from their responsibilities for a successful project. The two engineering managers influenced both the IT manager and area manager through their personal relationships with them. The engineering managers also advocated PICRS to the crew managers who directly reported to the area manager. Except for the developers, the members' ranks in KTP's organizational chart are roughly corresponding to their latitudes in this map.

The IT manager was initially against PICRS because the system was not corporate-developed while there was a Ford-wide policy that all IT projects should be coordinated at the corporate level. PFMR was a corporate-level system, but it did not work. The developers and the sponsors of PICRS project argued that PICRS could be regarded as a direct replacement of the old Excel-based system. Since the budget for PICRS was null, the project did not necessarily need to go through IT department's approval process. Our goal was to gain an OK from the IT manager, not his whole-hearted commitment. We succeeded through the "inventive" definition of the system, and the two engineering managers' relationship with the IT manager.

Two other very important stakeholder groups are the crew managers and the line supervisors. After we secured the commitment from the area manager, we simultaneously advocated the system to the crew managers and let the area manager influence her subordinates. Excited by the potential benefits PICRS could bring, the crew managers shortly became very enthusiastic about the project.

The line supervisors were the primary users of the system and their participation and support were extremely important. Initially, the supervisors got involved through the mandate from their direct reports, the crew managers. On the other hand, we also worked most closely with the line supervisors. A quality product developed based on the supervisors' direct input, our work relationships with them, and a mandate from the crew managers all contributed to the increase in the level of commitment of the line supervisors. They not only bought in to the system, but also contributed to make the implementation a success.

4.3.3 Notes on Communication

The importance of effective communication has been extensively discussed in matters such as change management. Federal Express CEO Fred Smith recently spoke to Fortune magazine, "To be able to change effectively, you have to have a high degree of trust and outstanding communications capability"[13]. Mr. Smith's comments resonate with my internship experience. Although the change I have created at Ford is far smaller in scale than Mr. Smith has done with FedEx, it can provide some similar insights.

Effective communication requires good understanding of both self and others. As Sun Tsu put it more than two thousand years ago in his famous work *The Art of War*, "If you know the enemy and know yourself, you need not fear the result of a hundred battles. If you know yourself but not the enemy, for every victory gained you will also suffer a defeat. If you know neither the enemy nor yourself, you will succumb in every battle.[14]" In my understanding, "know yourself" means know your stance, perspectives, assumptions, and capabilities. The same applies to "know the enemy". In non-confrontational situations, knowing both you and others is also one of the most important ingredients for any recipe of success.

Coming back to the PICRS project, we, after spending more than a month studying the situation and investigating potential solutions, got a good grasp of our situation, and what we were up to. By building a small test program, we knew, although was not 100% sure of, what we could do given the resources available. Another important thing was that we knew we had support from the management but no real pressure on our shoulders. We went all out to create the change, while as interns, had little to worry about.

We got to know others by proactively communicating with them. We talked with all the stakeholders on individual basis to understand their perspectives, concerns, and requirements. While actively developing personal relationships with all the stakeholders, we also paid attention to the hierarchical as well as informal networks at KTP's final assembly area through organizational and political lenses.

Understanding and communication are mutually-reinforcing. Good understanding contributes to effective communication, while adequate communication enhances mutual understanding. The reinforcing loop fosters mutual trust which is a requisite for successful cooperation (See figure 17).

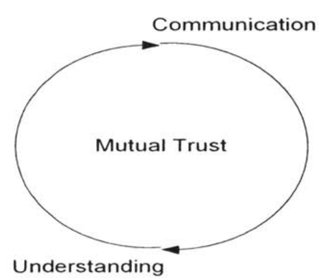


Figure 17: The communication model

Effective communication also requires consistency and patience. To deal with different people is an art. I felt this strongly throughout my internship project. Sometimes it was just too difficult to break the ice when the person I talked to was reserved and defensive. What I have found was that patience and consistency will finally pay off, while impatience, presumptuousness, condescension, and cynicism will always do great harm.

One example can illustrate why I think patience and consistency are important in effective communication. The first time I approached a superintendent about the implementation of PICRS on his line, he did not even talk to me. He was very busy on the floor at that time and basically ignored my existence. I felt irritated, but I also knew the attitude would not help. I approached him again after lunch. He asked me to wait and never came back to where I waited. So I thought maybe I really should not try to catch him on the line (superintendents have offices). I went to him a third time, and this time to his office. He appeared to be rather talkative. After chatting with him about Boston Celtics (He is about 6.3 feet tall and used to be a good basketball player) I demonstrated the system to him and his officemate—another superintendent I happened to know during a training session. They immediately became very interested in the system. Later they became two of the strongest advocators of the system. Had I lost patience in approaching that superintendent, I would have lost an important source of support.

4.3.4 Notes on Corporate Culture

The corporate culture is a big framework within which the internship project is carried out. Ford Motor Company has had a hierarchical organizational structure, and a culture where the decisions are usually passed from top down. In this culture, employees have fewer incentives to make changes. Even when there are innovative ideas from some lower level employees, the voice is difficult to reach the layer where people can make a go/no-go decision.

The PICRS project was done in such an environment. Instead of trying to challenge the culture (which would have certainly amounted to little gain), we, at the beginning, decided to carry out plans adaptive to the top-down culture. That is why we focused on securing management support at the very outset of the project. That is also one of the reasons why we chose the least-cost development method so that no budget decision needed to be made from the upper level management. We knew that the decision making procedures in such an organization tend to be lengthy, so we tried to avoid asking the bosses for approvals of particular actions. Fortunately, the supervisor and sponsor of the project granted us great autonomy, so that we could make our own decisions, such as when to implement the system on a certain line.

At the same time, we tried to cultivate a bottom-up culture in the small circles where we worked. We engaged line supervisors early on, and encouraged them to provide feedback. We upgraded the system with the supervisors and made them feel that their opinions were highly valued. We sent out emails and questionnaires to the line supervisors and engineers and tried to make them actively involved in the development and implementation process.

It also should be acknowledged that KTP, as one of the Ford's plants that have better management-union relationships, is trying to cultivate a plant-wise culture that encourages active involvement and innovation from floor-level employees. This change is in line with Ford's initiative to drive toward a lean behavior. The author witnessed, during the period of his internship, a growing willingness of operators and production supervisors to participate in value-creating projects. Without this general willingness to take part and the rapport between labor and management, the project would not have been a success.

Shop floor operators and supervisors disclosed to the author that they would actually love to participate in projects which they believed could create value for the company and eventually benefit them. Those people are the experts in their areas on the floor, so they often had better ideas of the effectiveness of floor-level projects than the implementers and managers that sponsored the projects. Projects that failed to achieve adequate effectiveness, such as PFMR, have had strong negative influence on the shop floor supervisors. Some of them became cynical about the usefulness of other projects. Therefore it is extremely important that any project should really create value for the company and incentives for the employees involved. KTP now has a growing sense of involvement from the employees, what it needs to specially focus on is to ensure the quality of projects that will require participation from the floor level employees.

Chapter Five: Benefits and Impact

For a profit-seeking organization such as Ford, the real benefit of any project will ultimately translate into the positive net present value (NPV). The goal of PICRS project is to provide better production information in a more efficient way. It has an indirect impact on KTP's profitability. Thus, the NPV of PICRS project is very difficult, if not impossible, to calculate.

Measurements, such as the increase in jobs-per-hour, and decrease in average line downtime could also be used to evaluate the effectiveness of PICRS. However, since PICRS' impact on the bottom line also tends to be gradual and long-term, and many other factors also affect line efficiencies, using line efficiency changes as measurements would hardly produce any clear results in a short run.

Another thing to keep in mind in doing cost-benefit analysis for PICRS is that there was no investment in either software or hardware. Since the cost was essentially zero (except for the time and labor of the two interns, which Ford did actually pay for), any potential economic benefit would be net gain.

This chapter discusses the benefits and impact that PICRS has brought and will bring in such a context: while some results are quantifiable, most are not. We try to provide an objective account, hoping to intrigue the readers to think about how such systems like PICRS can affect the bottom line of manufacturing organizations.

5.1 The User Feedback

After the PICRS had been up and running on all five production lines for more than a month, on November 18, 2003, we sent out a survey to collect user feedback from all of the 50 or so users of the system. The survey includes the following questions:

1. What is your general opinion about PICRS? Is it an improvement compared to the old Excel-based system?
2. Does PICRS save time for you in either data entry (for line supervisors) or report generating (for engineers, superintendents, and crew managers)? If it saves time, please estimate how much time it saves for you on a daily or weekly basis. If it does not save time or is even more time consuming, please feel free to say so.
3. Do you think PICRS provides better downtime information than the spreadsheets did? Is the information now more or less accurate, easier or harder to use?
4. What do you think is good and what is bad about the new system? Is there anything that PICRS can not do while the spreadsheet can?
5. Has the new system affected the way you react to the downtimes? If so, how?
6. Please rank the usefulness of PICRS using a number from 1 through 10. (10 means very useful)
7. Any other comments?

In the next several days, we received replies from ten users. Seven of the repliers answered those questions pretty thoroughly. Although the turn-out rate was quite low, the users who replied represented all the user groups. The overall feedback was highly positive. The PICRS system got an average score of 8 out of 10 in terms of usefulness. All the responses considered PICRS as a definite improvement over the old Excel-based system. Among the users, the managers, engineers, and supervisors had somewhat different comments on the system.

The managers and the engineers gave especially high remarks to the system. They all concurred that the system had provided them with better information and make their work easier. The managers suggested the expansion of the system into other areas of KTP, and the engineers provided some suggestions on the improvement of the system.

The supervisors had a somewhat mixed feeling about the system. While they agreed that the system was an improvement, some of them mentioned the longer time it took to input the downtime data. Limited downtime codes of frame lines as a drawback was also mentioned. At the same time, all the supervisors who gave feedback thought that the automatic generation of daily summary page was a good feature that had saved them significant amount of time daily.

Below are some quotes from the responses we received. It should be noted that, based on the feedback from the users, we upgraded the system and solved some problems (especially the problems with frame downtime codes) mentioned here.

"Yes - I think it saves more time and provides better tracking over time."

---Area manager's response to Question 1

"Logging downtime explanations take longer using the pull down menus as opposed to the old method of typing them all in. The production reports are good because of the combination of all entries in the summary which saves time."

---A production supervisor's response to Question 1

"Improvement but still not very good. Many operations are missing. I don't have much to choose from on a few teams and find myself having to make stuff up because I can't pick what I need."

---A production supervisor (frame line)'s response to Question 1

"It does save my maintenance supervisors and myself (MPS) time over the old way. I would estimate approx 20~30 minutes a day savings."

---A maintenance engineer's response to Question 2

"Yes - More detail. The maintenance organization now answers to more accurate problem descriptions provided by production."

---A maintenance engineer's response to Question 3

"We have more of an immediate feedback system and can see if something is getting worse, I particularly like the summary pages for downtime meetings."

--Area manager's response to Question 5

"If the downtime description is entered accurately then time wasted physically collecting facts from individuals can be greatly reduced. It allows us to go right to the source of the problem and get information first hand - (if it was entered accurately)"

--An engineer's response to Question 5

5.2 Impact on the Supervisors

To understand PICRS' impact on how the final area line supervisors work, one first needs to know how they used to work with the old system. It goes like this. Basically, one supervisor on each line in the final assembly area had to fill out a number of spreadsheet during each shift. Figure 5 in section 3.2.1 of this thesis shows one of the most important spreadsheet templates that the supervisor had to fill out with the old system. The supervisor input the production number and downtime information at the end of each production hour. It usually only took a few minutes to input the information for one production hour. The actual time required depended on the number of downtimes occurred during that hour and each individual's typing speed.

At the end of each shift, a supervisor, usually the second supervisor on the same line (there are two supervisors for each shift for a particular line), had to compile the whole shift's input into a summary report (figure 18 shows an example of this summary report). Generating this report takes roughly 20 minutes. This daily report is totally based on the information entered hourly by the supervisors. PICRS, as a true database application, can generate this report automatically. Thus, this 20 minutes/shift spent on compiling downtime data is totally eliminated. Figure 4 in section 3.1.2 of this thesis shows a counterpart report automatically generated by PICRS.

Kentucky Truck Plant
Production Volume

Department: Chassis 1 — Date: 4-Jun-03

Daily Summary

Line Hours:	10.0
Total Frame 1 Production:	331

	Units	Hours	JPH	Comments
Target	360	9.0	40.0	
Actual	331	9.0	36.8	
(Loss)/Gain to Target	(29)		(3.2)	

OEE 91.9%

Break (min):	60
Meetings (min):	0

Maintenance Explanation of Losses			Production Explanation of Losses		
Units	Description	Preventive Action	Units	Description	Preventive Action
8	Exc caught in crossover		14	Gaps from frame	
1	D8 Unchain sensor		2	Frame Flip	
0.5	E-5 Tire secure gun and rear bumper gun		1.5	EOL N/S	
0.5	Seat return elevator		0.5	Box Mix	
			0.5	D-28 FF OT	
			0.5	E-2 Repair	
			0.5	D-4 Tires out of sequence	
			0.5	E-25 Seats	
			0.5	D-17 Tedlar hung on unit	
			0.5	D-15 Brake Fill OT	
			0.5	E-16 Radiator Install	
			0.5	D-22 PIT repair	
			0.5	D-17 OD Fault	
			0.5	D-7 Isolator repair	
10	**Total**		23	**Total**	

Figure 18: Sample daily report in an Excel template (information in this figure is fictitious for confidentiality reasons)

As indicated in the previous section, the line supervisors actually had mixed opinions about PICRS. While they all agreed that PICRS was an improvement over the old system, some of them complained about the longer time it took to use PICRS to input downtime information. Several reasons attribute to this. First, with PICRS the supervisors have to input more information, such as *team number* and *downtime category*, while they only needed to write a short description with the old system. The additional input makes the downtime information more structured and detailed, although it requires extra time for entering—even if it is just one click or key stroke. Second, the supervisors are still learning to get used to the new system. For less computer-literate users, it takes slightly longer time to use the new system than they did with the old one. Some feedbacks also showed that, for more computer-literate supervisors, it actually takes less time to use the new system.

Another issue, which is also reflected in the survey results, is the question whether the new downtime codes actually provide better information than the old short descriptions. Several points should be stressed in answering this question. First, the supervisor input form of PICRS does provide a field called "comment" which can record whatever descriptions the user wants to enter to supplement the downtime entry. However, this "comment" information is not included in any report that provides aggregated downtime information (by aggregated downtime information, we mean information that is grouped using one of the fields in the record). It is included in the detailed report for maintenance engineers to help them better diagnose equipment problems. Entering comments on downtimes into PICRS takes considerable amount of time. Some supervisors choose to enter comments more often than others. This might be why some supervisors think using PICRS is more time consuming. Second, the quality of the information provided by PICRS largely depends on the design of downtime codes, which is independent on the structure of PICRS. In other words, to improve the quality of the information provided by PICRS, the main thing to do is to improve the downtime code list without having to change any structural part of PICRS. For example, it was found that the first list of downtime codes for frame lines lacked a lot of necessary details. We upgraded the list promptly and improved PICRS' usefulness on frame lines greatly.

Another concern is that, with PICRS, the supervisors would tend to think rather mechanically—trying to look for a downtime reason in the list rather than coming up with his own description. This is a legitimate concern. The evidence and potential effect of this is yet to be seen. It, however, should be noted that the vast majority of the downtime reasons are reoccurring ones. Even with the old system, the descriptions of most downtimes are somewhat codified.

PICRS also has one effect that the old system lacks. It leaves supervisors less room to play with the production numbers and also checks if there is any conflict in the data input. Job number matching functions and various warnings to remind the supervisors to account for all units lost help to make sure that the input is accurate and consistent.

5.3 Benefits to the Managers and Engineers

In chapter 3, some of the benefits PICRS brings to KTP final area managers and engineers have already been mentioned. This section mainly talks about PICRS' impact on the way they work.

Managers interact with PICRS only for generating and viewing reports. To them, PICRS has no downside. With PICRS, the managers now can view all kinds of customized report at anytime, a luxury they did not expect to get with the old system. Our interviews with the area manager and two crew managers produced very encouraging results. One crew manager said that now he often "plays with" the system to see the downtime pareto charts of various time ranges to get a feel of how the lines are performing. Before PICRS, he had to spend at least half a day in front of a PC to get data for an OEE (overall equipment effectiveness) meeting, now he only needs to make a few mouse clicks to get what he wants. Moreover, he now can get the information for any time range and any downtime he cares about, not just for a calendar month or week.

Some production engineers now rely on PICRS as an easy-to-use source of production information. They turn to PICRS when they need to write reports that need downtime data. Actually we have programmed several modules that can help engineers to retrieve certain information from the database and automatically generate reports in the formats they want. The maintenance engineers are now using PICRS to view downtimes entered by production supervisors, take actions if necessary, and provide feedbacks in a more timely manner. Some maintenance engineers are also trying to link PICRS to other information systems so that more comprehensive information can be provided.

5.4 Long Term Impact

Before PICRS all the information about production and downtime data is scattered in hundreds of Excel files and paper reports. To sort out some useful information from these files and reports was often a cumbersome task.

Meetings on equipment effectiveness were regularly held (weekly, monthly). Before each meeting, considerable effort had to be spent to create relevant reports for the managers and engineers to use. All this resulted in a phenomenon we call *Periodicity of production information*—only weekly or monthly information was available, not to mention the accuracy of such information. Ad hoc reports were a luxury which required considerable extra work.

PICRS is one step toward *continuous production information*, information that is stored in a central database where all the data elements can combine to generate needed information easily.

The long term impact of PICRS is only a subject of speculation. With continuous production information available, people may change their *periodic* behavior in treating line efficiencies and adjust to the *continuous* mode. This change may help them to have a better understanding of the status of their production lines and solve problems in a more timely fashion, thus in a long run increase line utilization and product quality.

PICRS was implemented onto the chassis lines on August 11, 2003. We use January 13(the first day that the data is available in the system) - August 10, and August 11 - December 4 (the last day of the project) as two periods to calculate the changes in average job-per-hour (JPH) for both lines. We find that for the post-implementation period, the average JPH of line 1 is 1.35 higher than that of the before-implementation period. For line two, the change is 0.15 increase in JPH. In the calculation, we took out the outliers—days that have extremely low JPH because of some unusual machine breakdowns or construction projects—to make the results more meaningful. We, however, understand that changes in such small scale on a time base of several months can well be the result of random fluctuation and can also be due to many other factors. It needs more investigation to know the change in line utilization due to the use of PICRS.

5.5 Conclusions and Future Work

In conclusion, PICRS was a system that we successfully developed and implemented within the six months of the internship time frame. From all the feedback we have got so far, we feel comfortable to say that PICRS is an improvement over the old Excel-based information system. It has eliminated some of the time consuming work that the supervisors and engineers had to do with the old system. It can also provide more structured and accurate information in a more efficient manner. The relational database of PICRS is also capable of providing various useful information that well-designed queries can generate. Carefully-designed interfaces make the user interaction with the system easy and provide ample options for report generation.

The quality of the information provided by PICRS is largely dependent upon the design of the downtime codes. Codification of downtime reasons are necessary for the functioning of database queries, which are one of the most powerful tools of relational databases. However, the superiority of codified downtime reasons over the old way of unstructured descriptions is arguable.

Long term impact of PICRS on KTP final assembly area is yet to be seen. We speculate that PICRS can help managers and engineers change their behavior of managing line efficiencies from the *periodic mode* to the *continuous mode*, which might help them to solve downtimes in a more timely manner.

As already suggested by some users, future work can be done to expand PICRS to other production areas at KTP. As the system grows, the database can be migrated to a more powerful platform such as Oracle. Work can also be done to link PICRS to other production information systems currently running at KTP. The integrated system can link relevant data together and provide more useful information.

Bibliography

[1] *Ford's Troubles*, Economist, 2003 6/14-6/20, page 60, source: automotive news

[2] *Extinction of the car giants*, Economist, 2003 6/14-6/20, page 11

[3] Ford Motor Company, *2002 Annual Report*

[4] Iivari, J., Hirschheim, R., and Klein H.K., *Beyond Methodologies: Keeping up with Information Systems Development Approaches Through Dynamic Classification*; proceedings of the 32nd Hawaii International Conference on System Science, 1999

[5] Huang, Y., *Prototyping : Throwaway or Evolutionary?*, http://www.scis.nova.edu/~yaochin/660-p.htm, 1998

[6] Seaman, C., *COTS-based Development Process in the FDD*, http://sel.gsfc.nasa.gov/website/documents/study-br/cots.htm, 11/1997

[7] whatis.com database definitions, http://searchdatabase.techtarget.com/sDefinition/0,,sid13_gci211895,00.html, 2/22/2004

[8] *Introduction to data modeling*, UT Austin Information Technology Service, http://www.utexas.edu/its/windows/database/datamodeling/rm/rm4.html, 2/29/2004

[9] Chen, P.P., *The Entity-Relationship Model - Toward a Unified View of Data*, ACM Trans. Database Syst. 1(1): 9-36, 1976

[10] Ancona, D.G., Kochan, T., Scully, M., Van Maanen, J., and Westney, E., *Organizational Behavior & Processes: Managing for the Future*, South-Western College Publishing, 1999

[11] *Sloan Leadership Model*, http://sloanleadership.mit.edu/leadership.pdf, 2004

[12] Beckhard, R., Harris, R., *Organization Transitions*, Addison Wesley, 1987

[13] *CEOs on Innovation*, Fortune, vol. 149, NO. 5, 3/8/2004

[14] Sun Tsu, Translated by Griffith, S.B., *The Art of War*, Oxford Press, 1971

5044-70