Statistical Process Control Charts for the Computer-Aided Fabrication Environment

by

Nicole S. Love

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1994

Author ....
Department of Electrical Engineering and Computer Science
May 11, 1994

Certified by ....
Donald E. Troxel
Professor of Electrical Engineering
Thesis Supervisor

Certified by ....
Duane S. Boning
Assistant Professor of Electrical Engineering
Thesis Supervisor

Accepted by ....
Frederic R. Morgenthaler
Chairman, Departmental Committee on Graduate Students

Eng.

Statistical Process Control Charts for the Computer-Aided Fabrication Environment

by

Nicole S. Love

Submitted to the Department of Electrical Engineering and Computer Science
on May 11, 1994, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering

Abstract

Statistical Process Control (SPC) charts are a means of improving process performance through the analysis of post-process measurements. SPC charts for the Computer-Aided Fabrication Environment were developed to assist in the manufacturing of integrated circuits. This thesis describes the application program to produce the SPC charts focusing on four main objectives: generating the control charts, accessing the data from the database, caching the data, and combining control charts. The generation of the control chart is independent of the data and requires a data file of the correct format. The control chart user interface is implemented using the Tcl/Tk scripting language along with [incr Tcl], an extension to Tcl. Accessing the data is accomplished by using the Process Flow Representation to link a process step with the corresponding inspection step. Caching the data is accomplished using a Gestalt object to store the time frame of the data, the data, and other information used in generating the control chart. The advantages and disadvantages of combining control charts, including multiple machines running the same recipe, and multiple recipes on the same machine, are examined.

Thesis Supervisor: Donald E. Troxel
Title: Professor of Electrical Engineering

Thesis Supervisor: Duane S. Boning
Title: Assistant Professor of Electrical Engineering

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Computer Integrated Manufacturing (CIM) is becoming an important component of manufacturing industries. CIM enables the integration of every aspect of manufacturing, factory automation, manufacturing planning and control, design, simulation, documentation, etc. MIT is developing a software system in the Computer Integrated Design & Manufacturing (CIDM) research project. A component of the system, CAFE (Computer Aided Fabrication Environment) has the goal of encompassing "all phases of process design, development, planning, and manufacturing of integrated circuits." [11]

As in any manufacturing environment, in order to achieve the desired quality of a product, the semiconductor fabrication process must produce the same product with a small variation. Therefore the constant or periodic monitoring of a process is a requirement. Achieving and maintaining the desired quality of a process requires that the products meet the desired target levels. Applications to provide analysis of post-process measurements to maintain quality control aid in the manufacturing of integrated circuits. The development of Statistical Process Control (SPC) charts in CAFE provides a basis for management decisions and helps trouble shooting within the facility. SPC charts present the operator with a clear view of the state of a process over a period of time.

The following chapters detail the implementation of SPC charts in CAFE. The generation of SPC charts entails several components.

- Accessing the data from the database using the CAFE data structure.

- Caching of the data to increase the speed of generating the control chart.

- Displaying the control chart via the graphical user interface.

- Combining charts to look for important process trends:

    1. Two control charts for processes operating on the same machine.

    2. The same process operating on two different machines.

Chaps. 2 and 3 describe SPC in general and SPC charts in particular, presenting the background for SPC charts and providing an illustration of the format. The software components on which the application was based are described in Chapter 4, and a detailed description of the graphical user interface is presented. Chapter 5 provides a description of the algorithm used to access the data from the Gestalt database. The caching scheme for the purpose of increasing the speed of SPC chart generation is discussed in Chapter 6, Chapter 7 describes the advantages and disadvantages of combining SPC charts.

Chapter 2

Statistical Process Control

The ultimate goal of SPC is to improve process performance. The measure of process performance in SPC is determined by the variation of the process output variable. Walter Shewhart at Bell Labs believed that processes fell within predictable limits of variation based on statistical methods and deviations from the limits signalled problems in the process [4]. The economic success of a process is dependent upon a process that is in statistical control. Shewhart suggests a process which is "out of control" is not economical. If the process is "out of control" the process is not behaving "as it was designed to run." [4, p. 124] Large variations in a process affect the quality of the product. Economically, SPC helps to produce a product efficiently which reduces the waste associated with a process and increases the productivity. Through the analysis of SPC, decisions affecting a process can be made due to the detection of possible problems in the process.

SPC is used to monitor and to aid in maintaining the performance of a process. SPC can provide an analysis to determine if the variations observed are within the known noise limits for the process; the noise limits are determined statistically by the history of a process. The decisions regarding the stability of a process are based on the information given by SPC charts. The use of SPC is mainly to reduce variability and to maintain the stability of a process. From the analysis of the data, SPC can help in determining if a variation is by chance or if a variation has an assignable cause.

Chance causes are natural variations of a process resulting in a small deviation from the average value of the data. Assignable causes (unnatural variations) are usually variations due to operator errors, defective raw material, problems with the machinery, etc. Assignable causes normally produce large variations which lead to a process that is "out of control." "Out of control" is a term meaning the output is not conforming to a normal distribution as expected. Detection of assignable causes is important in maintaining control of a process and is essentially the focus of SPC charts. SPC charts use post-process measurements to recognize and detect major variations and potential instabilities of a process.

The purpose of the CAFE system is to manage integrated circuit fabrication facilities. The previous method of constructing a control chart consisted of manually extracting the data from a report of the process and graphing the control charts by hand. Thus the CAFE system needs a tool which will generate SPC charts. The addition of an application to produce SPC charts will aid in trouble shooting by the operator. By alerting operators to possible problems when the SPC charts are generated, the SPC charts will make trends in data apparent and aid in the prevention of possible problems in a process.

2.1 Product vs Process Control

Product control is an attempt to insure that the final output is 100% defect free. Fig. 2-1 demonstrates the product control method of quality control [4]. Product control entails 100% inspection of the product and determines if the product is defective. The defective products are then reworked for shipment or scrapped. Process control, unlike product control, aids in improving the process to reduce the defects produced. The process control model in Fig. 2-2 is a feedback loop designed to improve a process based on the analysis of the post-process measurements.

In product control, the main objective is to produce a set amount of products in a given time frame. To achieve this objective more products are produced to compensate for the defective products by the addition of longer hours or additional machines

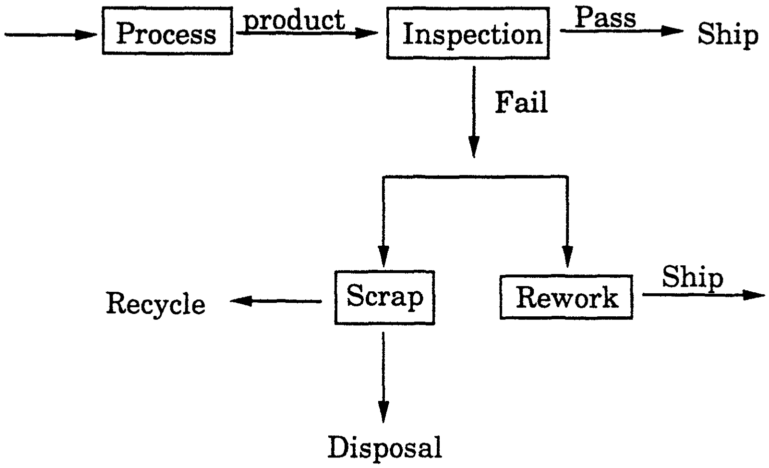


Figure 2-1: Product Control Model

to meet the deadlines. Product control methods make no effort to increase the efficiency of the process which produces the defective products, but process control does. Process control focuses on correcting the causes of defects, thereby increasing the efficiency of the process. SPC charts provide the evaluation of the data to determine if the process is in statistical control.

2.2 Run by Run Process Control

The importance of SPC is also seen in run by run process control which combines SPC and feedback control [17]. Run by run process control is a means of improving new processes by monitoring and analyzing post-process measurements after each run. Fig. 2-3 shows a diagram of the MIT Run by Run Controller. The Run by Run Controller uses SPC to determine if a process is drifting from the target level (gradual mode) or if a process has made a large shift from the target level (rapid mode). Run by run process control after each run uses the analysis from SPC charts

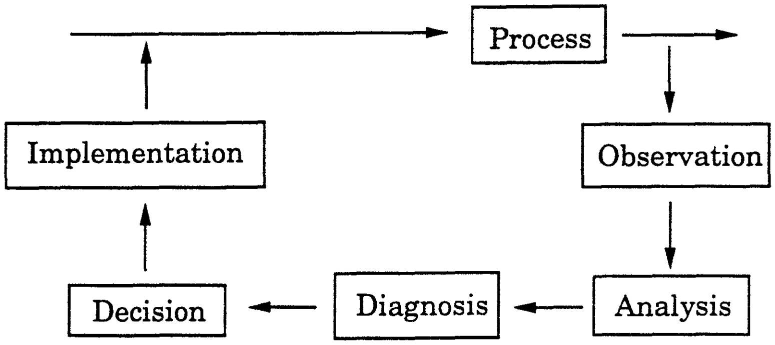


Figure 2-2: Process Control Model

to determine if the process should be altered via feedback control. The gradual and rapid modes are the feedback control mechanisms used to modify the process in order to maintain statistical control. The modifications made by the gradual and rapid modes are attempts to keep the output as close to the target level as possible (and not just within the same specification limit).

SPC charts determine if the process is "out of control" and feedback control changes the process to provide the desired output. The run by run controller cycles through this process of analyzing and altering the process to obtain the target levels set for the process. Due to the constant manipulation of the input parameters an accurate analysis from the SPC charts may only be obtained through the alteration of previous measurements. Compensations to the previous data must be made to reflect the changes made to the process due to a drift or shift from the target level, thus the term "Generalized SPC."

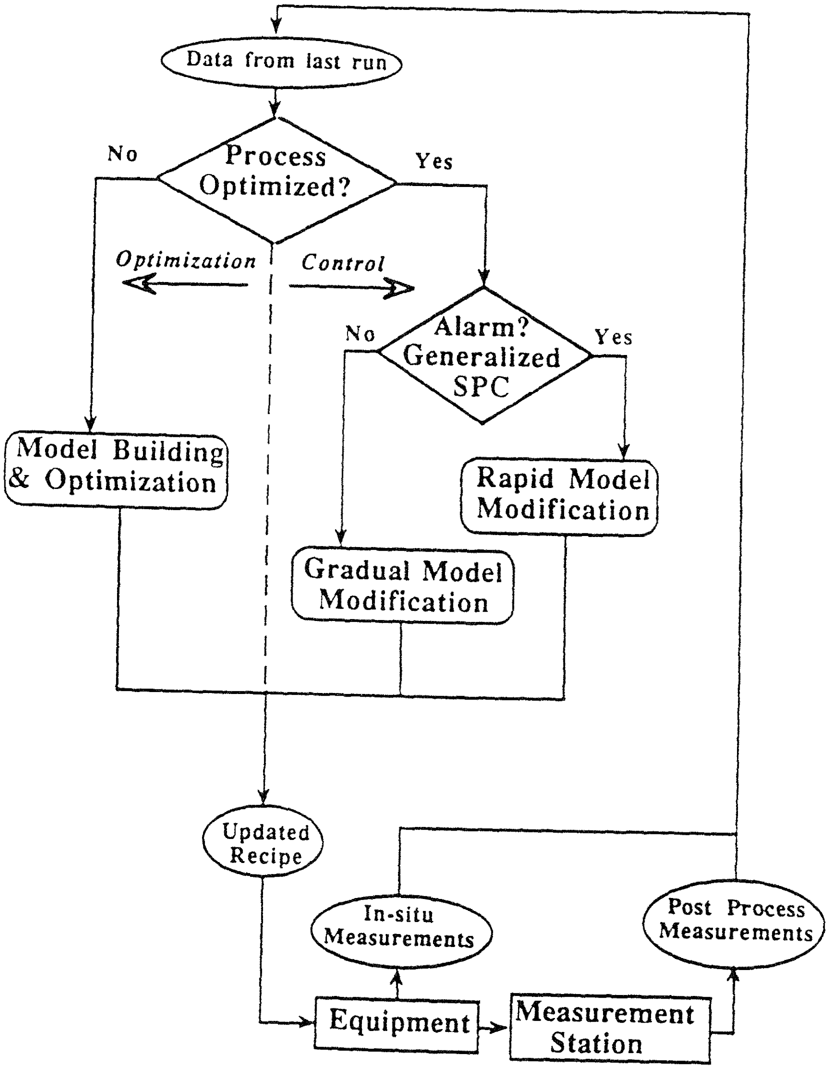


Figure 2-3: Run by Run Controller Flowchart

Chapter 3

SPC Charts

SPC charts have been a key component in manufacturing since Shewhart's design of control charts. Control charts continue to provide managers a measurement of progress, engineers a basis for design specifications, and operators a trouble shooting tool.

Control charts are a graphical representation of process state via plots of post-process measurements. The control charts also contain upper and lower limits as well as a central line and warnings of possible instabilities in a process. The central line is the average of the data, providing a basis to compare measurements. The control limits are determined from the data, but the specification limits are usually an estimation or a desired limit for a process. The post-process measurements are assumed to form a normal distribution centered around the average. The upper and lower control limits are 3σ above and below the average measurement.

There are several types of control charts available for analyzing post-process data; the most common are the range (R-chart), standard deviation (s-chart) and the average or mean ($\overline{X}$-chart) charts. The R-chart is used to study the range of a set of sample measurements, which is useful in determining the run to run variation in the spread of the measurement distribution. The s-chart also examines the spread of the distribution, in this case based on the standard deviation of sample measurements. The most common of the control charts is the $\overline{X}$-chart which was chosen to be implemented in CAFE. The $\overline{X}$-chart displays the variation from run to run in the average of the sample measurements.

The assignable causes affecting $\overline{X}$-charts fall into two categories: causes which directly affect, and those which indirectly affect the $\overline{X}$-charts. [3] Assignable causes directly affecting the $\overline{X}$-chart are causes which affect all of the products during a process. Temperature changes, differences in machine settings or the change in the operator are examples of causes which affect all the product in a process. Indirect assignable causes may be due to a "change in the proportion of distributions," single points outside the control limits caused by a wrong setting, or a measurement error.

The format of the charts is designed for ease of use, and to make any apparent problems clear to the lab technician. The format of the control chart is very similar to an acceptance control chart. Acceptance control charts are used for processes in which the importance is not focused on statistical stability of a process, but rather the importance lies in maintaining a process in which the output falls within acceptable (tolerance) levels. The acceptance control charts contain upper and lower tolerance levels, upper and lower control limits, along with upper and lower rejection levels. The region between the acceptance and rejection levels is deemed an indifference region. The format for the control charts in CAFE as seen in Fig. 3-1 consists of the control limits (3σ limits), the central line, the specification limits set by the user or obtained from the database, the target which is the desired average or goal of a process, and the step limits above and below the central line. Below is a description of the control chart components.

Control Limits. The upper and lower control limits are determined from the data. The calculation of the control limits is based on the assumption that the measurements form a normal distribution centered around the mean. Fig. 3-2 shows the probability distributions of the control chart. The spread of the distribution is determined from the average range of the sample measurements [3]. The following equations are used to calculate the control limits of the control chart. The constant A_2 is a factor developed for the calculation of the 3σ limits based on the sample size being used. The Table 3.1 contains a listing of A_2 and the corresponding sample size which is

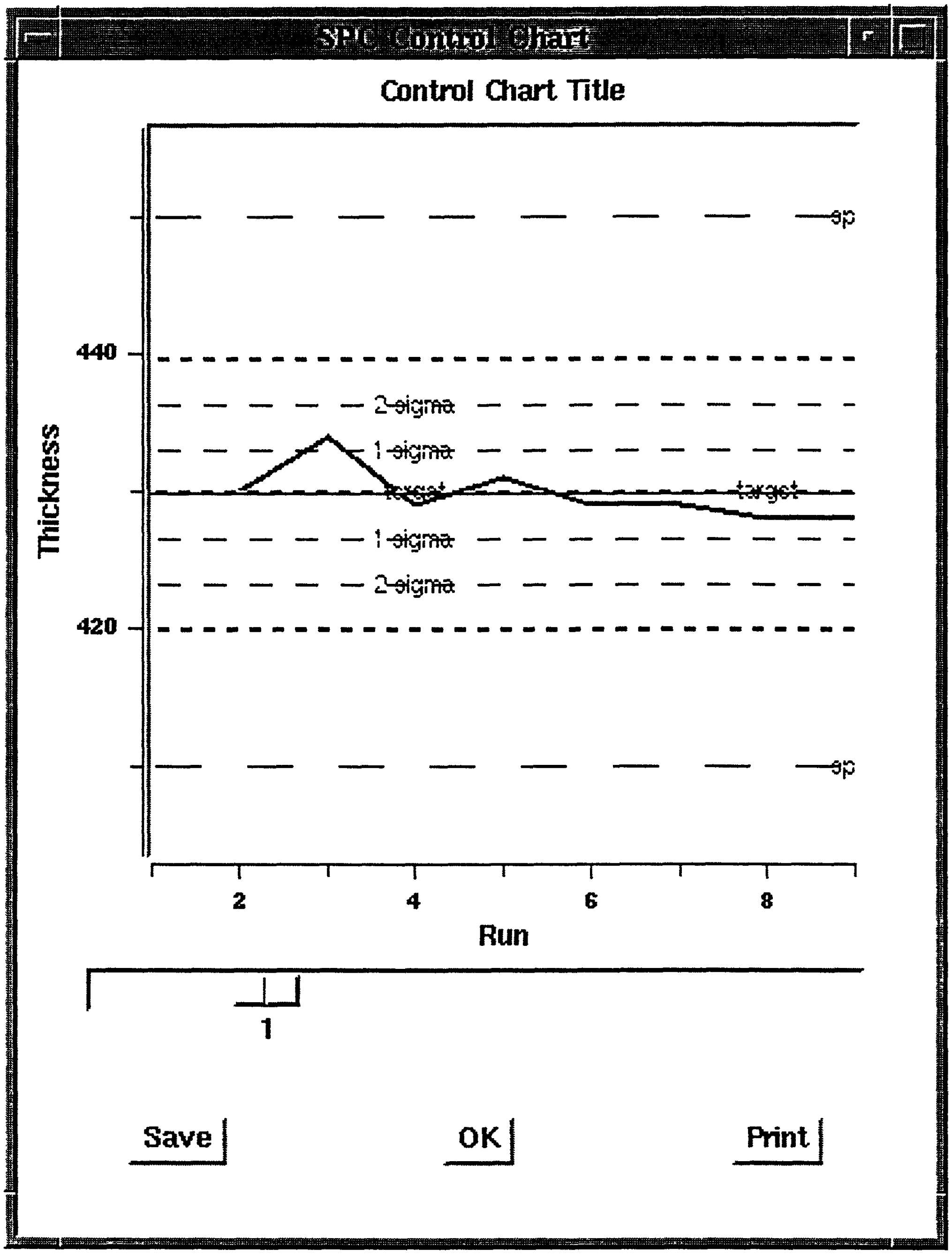


Figure 3-1: Example SPC chart

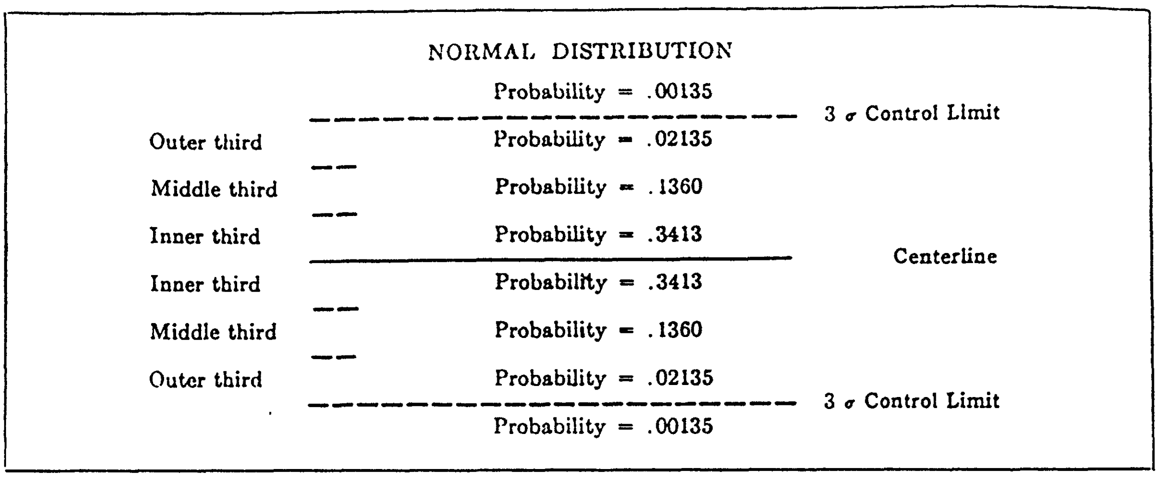


Figure 3-2: Probabilities associated with limits

used to calculate the 3σ value from the mean. The control limits establish the region of statistical stability of a process. The following equations are used to determine the upper control limit (UCL) and the lower control limit (LCL).

$$UCL = \overline{X} + A_2\overline{R}$$

$$LCL = \overline{X} - A_2\overline{R}$$

Sigma Limits. The sigma limits which partition the chart into 6 equal sections are used in detecting potential instabilities in the process based on the Statistical Quality Control Handbook by Western Electric Company. The rules which are discussed later help to determine if a process is in statistical control.

Center Line. The center line is the average of the sample measurements and is considered the center of the distribution of the measurements. The center line determined from the data is the solid line in the center of the chart, and may from the target, as is explained below.

Number of Observations in Subgroup/Sample	A_2 Factors
2	1.880
3	1.023
4	0.729
5	0.577
6	0.483
7	0.419
8	0.373
9	0.337
10	0.308
15	0.223
20	0.180

Table 3.1: A_2 Factors

Target. The target is the desired output of the process which may be estimated from the experience of the engineer or based on known output measurements of the process. The target is usually a goal set for the process. The difference between the target and the center line should not be great or alterations should be made to the process in an effort to achieve the desired goal.

Specification Limits. The specification limits as well may not be determined from the process, but may be the acceptable limits set by an engineer. The specification limits may be within or outside the control limits; it is of course more desirable for the specification limits to fall beyond the control limits. The specification limits may be more important to the operator than the stability of a process and as long as the output is within the specification limits and is not moving towards exceeding those limits, the process may be considered to be behaving adequately for the given process.

3.1 Detection of Instability

The detection of instability in a process makes the operator aware of potential problems, and is an important aspect of trouble shooting. The aid of control charts in trouble shooting for any process is economically beneficial. Being able to fix problems before the problem is "out of control" is always a goal in industry. Possible instabilities in a system are determined using the rules or criteria set by Western Electric Company [3]. The following is a listing of these criteria.

1. A single point beyond 3σ limit.
2. Two out of three successive points beyond 2σ limit.
3. Four out of five successive points beyond 1σ limit.
4. Eight successive points on one side of the central line.

The rules are based on the normal distribution of the measurements. Fig. 3-2 shows the probabilities associated with the regions of the control chart. The probability of meeting any of these criteria is described below.

Criterion 1. The probability of a point beyond the 3σ limit is

$$P = 0.00135.$$

Criterion 2. The probability of 2 out of 3 points beyond the 2σ limit is

$$P = 0.0227 \times 0.0227 \times 0.9773 \times 3 = 0.0015.$$

Criterion 3. The probability of 4 out of 5 points beyond 1σ limit is

$$P = 0.1587 \times 0.1587 \times 0.1587 \times 0.1587 \times 0.8413 \times 5 = 0.0027.$$

Criterion 4. The probability of 8 points on one side of the center line is

$$P = 0.5 \times 0.5 \times 0.5 \times 0.5 \times 0.5 \times 0.5 \times 0.5 \times 0.5 = 0.0039.$$

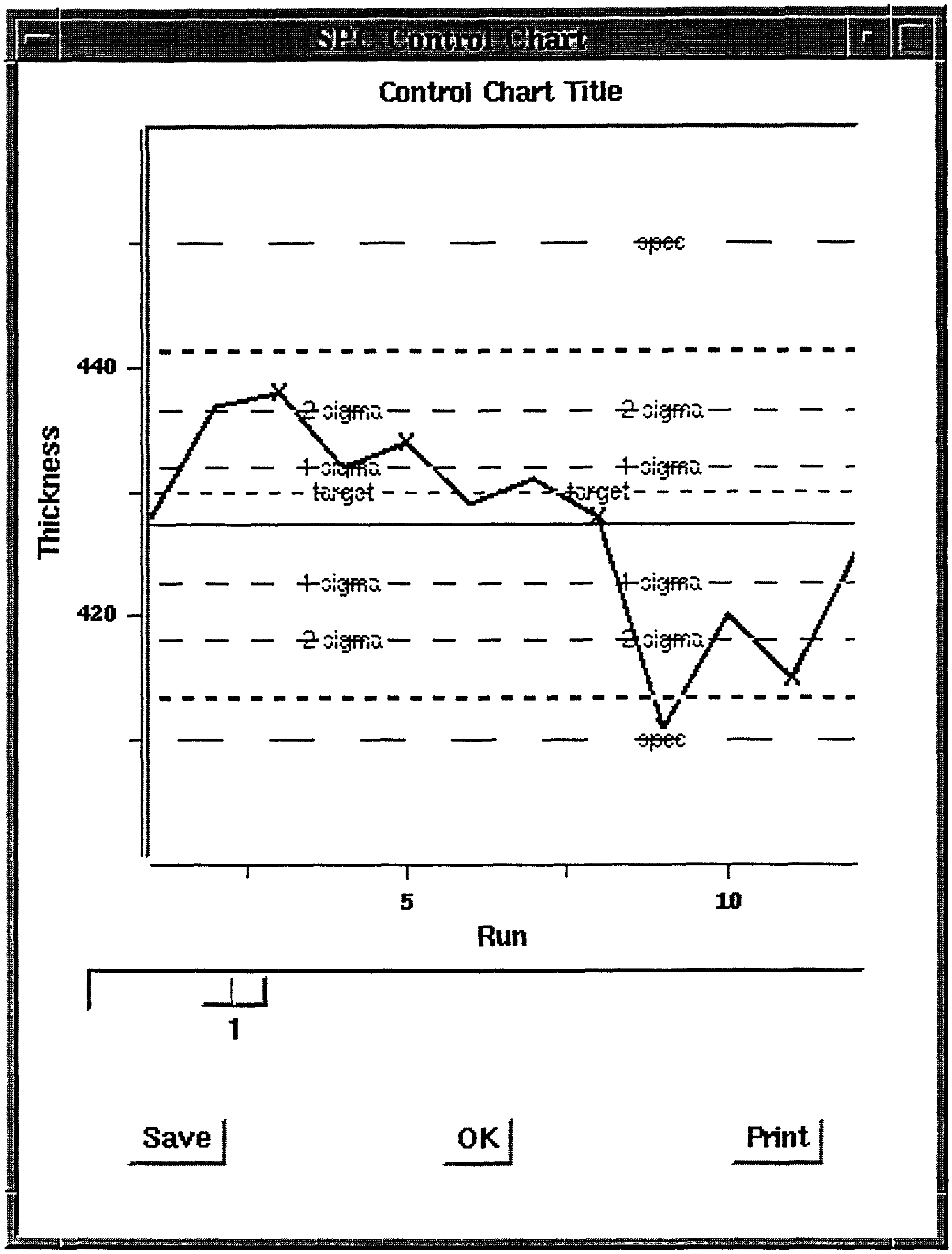


Figure 3-3: Example of Instability

Fig. 3-3 shows a control chart for a process that is "out of control." The "X" marks are the warnings indicating that one or more of the criteria have been met and indicate instability.

- The point associated with Run 9 meets Criterion 1.
- The point associated with Run 3 meets Criterion 2 due to the combination of Run 2 and Run 3.
- The point associated with Run 5 meets Criterion 3 due to the combination of Runs 2 to 5.
- The point associated with Run 8 meets Criterion 4 due to the combination or Runs 1 to 8.

These rules will determine if the process is "out of control", and serve to alert the operator to any potential problems with the process.

Chapter 4

Software

This chapter begins with an overview of the control chart software architecture, and the flow of control in running the charts. The underlying database and the implementation languages on top of which SPC charts were implemented are then described. Finally, the user interfaces for designing/editing and for displaying control charts are described in detail.

4.1 Architectural Overview

The application program to generate SPC charts in CAFE uses both Tcl/Tk and Common Lisp to implement the desired functionality. Fig. 4-1 displays the architectural overview of the application program. The generation of a SPC chart begins by designing the control chart. From the CAFE menu the **Design/Edit Chart** option is selected which calls the **design-chart** Tcl/Tk script file. The **design-chart** script file displays a form to be completed. When the name of the form is specified and the <Tab> or <Return> key is pressed, a chart will appear. If the chart was previously designed and the information was saved in the database, the chart will contain the necessary labels and the information will appear in the corresponding fields. If the <Tab> or <Return> key is pressed after a field the label of the chart will be changed to the appropriate value. Upon completion of the form and pressing the Save button the information is saved in the database. The details of the form are discussed in

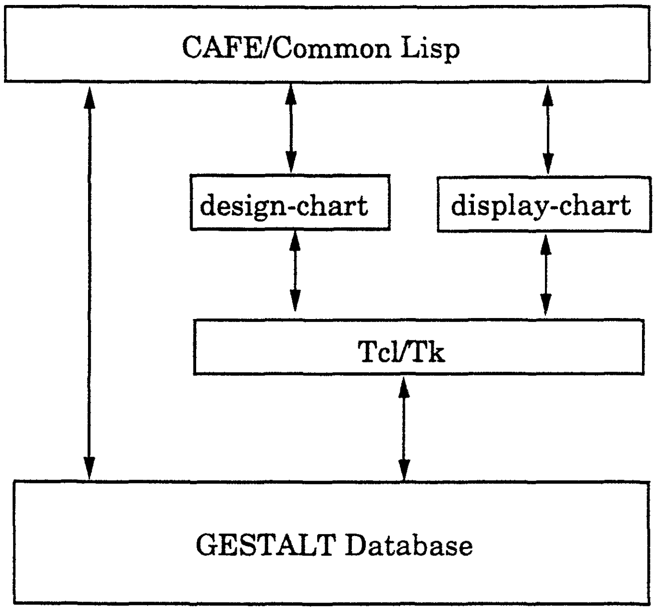


Figure 4-1: Architectural Overview

Sect. 4.4. The actual display of the chart occurs when the CAFE menu **Display Chart** option is selected. The name of the SPC chart is specified and the data is accessed based on the information given in the form. The **display-chart** Tcl/Tk script file is called with the name of the control chart and the file containing the data as arguments. The control chart is then displayed once the <Tab> or <Return> key is pressed.

The application program is partitioned into several Tcl/Tk script files and a Common Lisp file. The Tcl/Tk script files are divided based on functionality. Fig. 4-2 shows a diagram of the structure of the Tcl files and a description of the files follows.

chart.tk contains the control_chart class.

gen_chart.tk contains the procedures common to both the design and the display of the control chart, including procedures to specify the movement of the cursor.

design_chart.tk contains the internal procedures associated with the design of the control chart.

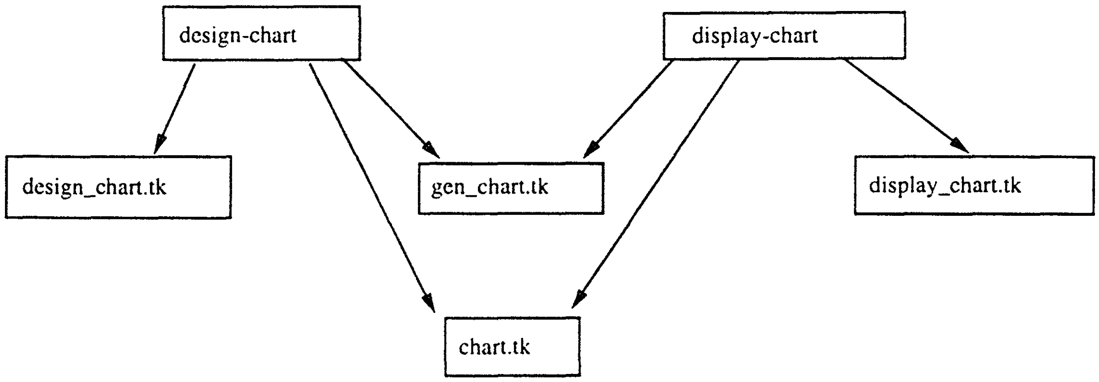


Figure 4-2: File structure (note: The arrows represent calls to procedures declared in the indicated files).

design-chart is an executable containing procedures to generate the widgets displayed on the terminal for the design form.

display_chart.tk contains the internal procedures associated with the display of the control chart.

display-chart is an executable containing procedures to generate the widgets displayed on the terminal for the display form.

The **`design-chart`** and **`display-chart`** files access procedures in the remaining files to display the charts and respective forms. The **`chart.tk`** file contains the **`control_chart class`** and may be sourced for other application programs requiring SPC charts. The **`gen_chart.tk`** contains procedures which are called by both the **`design-chart`** and **`display-chart`** files. Procedures to control the movement of the cursor, access the database, store data in the database, etc. are contained in the **`gen_chart.tk`** file.

4.2 GESTALT

GESTALT provides an object based interface to database systems. GESTALT is currently implemented in CAFE as a layer of abstraction linking application programs to database systems [9]. The GESTALT schema contains object types which may be either pre-defined or domain-specific and operations defined on the instances of the object types. Pre-defined object types such as INTEGER, FLOAT, etc. provide a basis for the creation of domain-specific object types. Domain-specific object types are defined to represent "objects in the application domain," such as process flows, machines, and measurements in our case (integrated circuit fabrication).

Operations in data abstractions can be divided into the following categories:
selectors - procedures to select an object component
mutators - procedures to update or delete an object
constructors - procedures to create objects
generators - procedures to generate null objects and iterators [9, p. 57]

A description of a database type and attribute from the GESTALT schema can be seen using the schema_dump program. The information provided is useful for application programmers. Fig. 4-3 shows a description of an example database object type (a "`trend`") in the schema. The application programmer can use this information to determine the output of a selector, the object type of a particular component, the allowable operations on an object type, etc.

4.3 Tcl and Extensions

4.3.1 Tcl/Tk

Tcl (Tool command language) is an interpreted scripting language, and Tk is a toolkit for X Window Systems providing commands for building interactive graphical user interfaces. Tk provides a large number of user interface classes, including scrollbars, scales, frames, menus, etc. New classes may be defined by the programmer in C. [incr Tcl] is a further extension to Tcl supporting object-oriented programming, and can

```
Schema Dump description of terms:
  1-1 -> Y - Corresponds to a single attribute value.
         N - Corresponds to a list of attribute values.
  Act-> Y - The attribute's value belongs to this object.
         N - The attribute's value belongs to another object.
  Inv -> Y - The object(s) can be retrieved given the attribute's value.
         N - The object(s) can't be retrieved given the attribute's value.
  Unq-> Y - Two objects can't have the same attribute value.
         N - The attribute's value can be duplicated.
  Nul-> Y - The attribute's value can be null.
         N - The attribute's value can not be null.
  Mut-> Y - The attribute's value can be changed.
         N - The attribute value is read only.
  Prf -> Y - The attribute's value is retrieved when the object is.
         N - The attribute's value is retrieved when it is accessed.

Name: TREND
   Description:       Type used in trend analysis
   Location:          INGRES
   Domain Specific?:  Y
   Deletable?:        Y
   Superclasses:
Attribute Name
   Attribute Type                 1-1 Unq Nul Inv Act Mut Prf Misc. Information
-------------------------------------------------------------------------------
name
   STRING                          Y   Y   N   Y   Y   N   Y size:64
owner
   LABUSER                         Y   N   N   N   Y   N   N
advice
   VARSTRING                       Y   N   Y   N   Y   Y   N size:128
processdescs
   PROCESSDESCRIPTION              N   N   Y   N   Y   Y   N
processflows
   PROCESSFLOW                     N   N   Y   N   Y   Y   N
readingsextractor
   VARSTRING                       Y   N   Y   N   Y   Y   N size:128
limithigh
   FLOAT                           Y   N   Y   N   Y   Y   N
limitlow
   FLOAT                           Y   N   Y   N   Y   Y   N
title
   VARSTRING                       Y   N   Y   N   Y   Y   N size:128
axislabel
   VARSTRING                       Y   N   Y   N   Y   Y   N size:128
timeframe
   TIMEINTERVAL                    Y   N   Y   N   Y   Y   N
```

Figure 4-3: Schema dump of a database object

be used to define additional classes for building user interfaces.

A Tcl/GESTALT interface was developed to allow Tcl programs access to the database [1]. Through the command `dbeval`, Tcl programs may perform any of the operations associated with selectors, mutators, constructors, and generators.

The user interface classes used in the control chart application are described below. Fig. 4-4 is an example containing the widget components.

Frames The frame widget is used to hold other widgets and provides for complex layout capabilities.

Blt_graphs The blt_graph widget provides basic XY plotting capabilities. The widget has numerous options to facilitate flexibility in viewing the desired data.

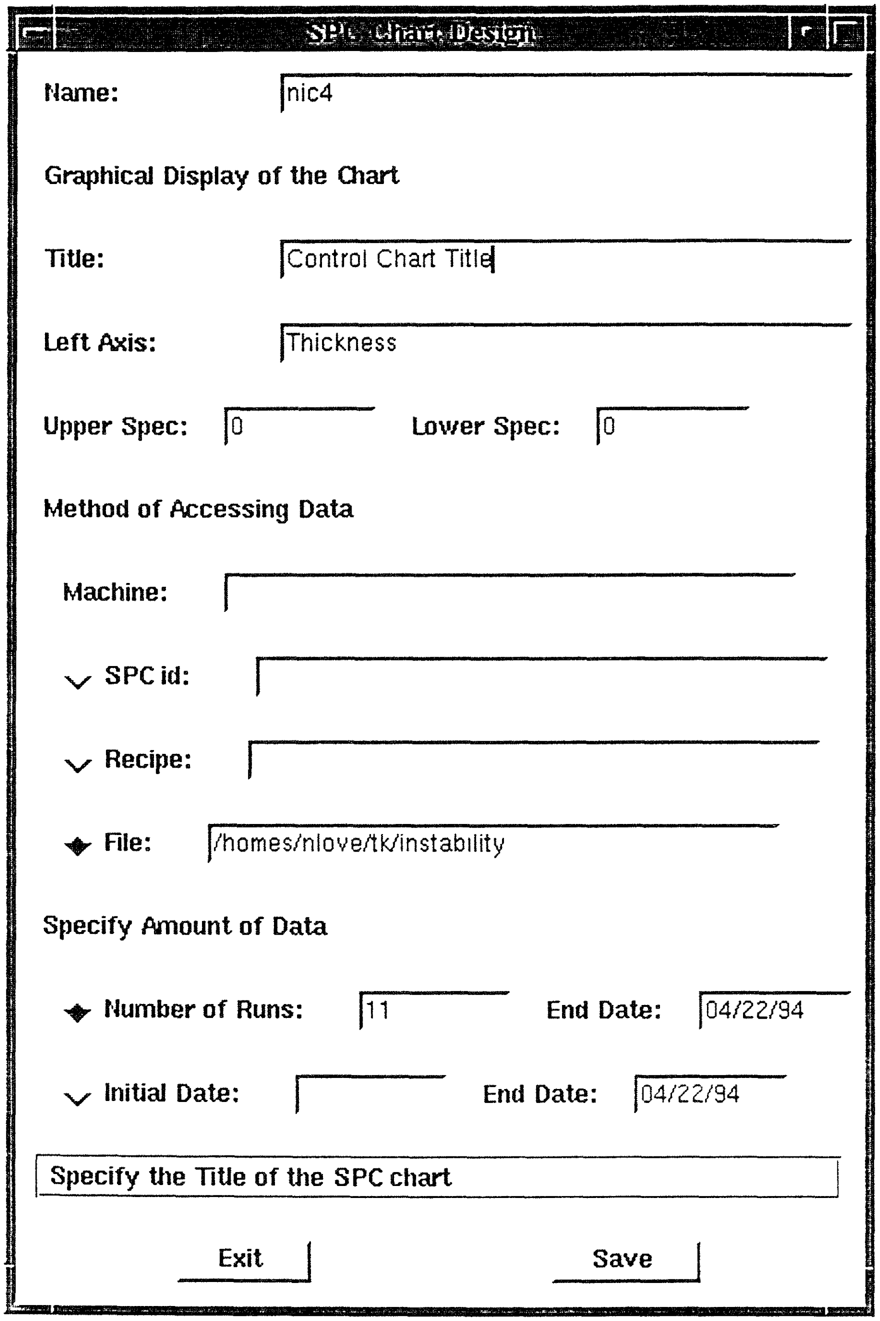
Buttons A button widget displays a text string; when the button is invoked by pressing and releasing a mouse button an associated command is invoked.

Scales A scale widget consists of a slider which has a value at each position. A specified command may be invoked each time the value of the scale is changed.

Radiobuttons A radiobutton widget is primarily used in groups to allow for a choice between different options. All radiobuttons with the same variable allow for only one radiobutton to be selected at any given time. The variable is set to a specified value when the radiobutton is selected.

Entries An entry widget contains a text string which can be edited. A variable is automatically set to the value of the entry for later use in the program.

Labels A label widget contains a text string which can be associated with a variable to allow the label to be altered anywhere in the program.



Name: nic4

Graphical Display of the Chart

Title: Control Chart Title

Left Axis: Thickness

Upper Spec: 0 Lower Spec: 0

Method of Accessing Data

Machine:

◇ SPC id:

◇ Recipe:

◆ File: /homes/nlove/tk/instability

Specify Amount of Data

◆ Number of Runs: 11 End Date: 04/22/94

◇ Initial Date: End Date: 04/22/94

Specify the Title of the SPC chart

Figure 4-4: Control Chart Design/Edit Form

4.3.2 incr Tcl

[incr Tcl] is an object-oriented extension to Tcl which aids in organizing procedures and data [12]. [incr Tcl] was used in this thesis work to create a new control_chart class. The new class is a template for objects which can be created. The control_chart class gives Tcl application programs the ability to create control_chart widgets separate from the SPC chart application specific to CAFE. The following shows the syntax and description of a class definition; key elements of class definitions are explained below.

```
itcl_class className {
inherit baseClass ?baseClass?  ?...
    constructor args body
    destructor body
    method name args body
    proc name args body
    public varName ?init?  ?config?
    protected varName ?init?
    common varName ?init?
}
```

Inherit The new class `className` will inherit the characteristics of the base classes specified.

Constructor The constructor is invoked upon creation of an instance of the class.

Destructor The destructor is invoked when an instance of the class is deleted.

Method Method is used to declare a method with the name `name`. A method is similar to a procedure, except that a method has access to public, protected and common variables in the class definition.

Proc Proc is used to declare a procedure such procedures have access only to common variables in the class definition.

Public Public variables can be modified by any method (or using the "config" method). A command can be specified to be invoked whenever a public variable is modified using the "config" method.

Protected Protected variables may only be modified through methods and indirectly by procedures.

Common Common variables are shared by all instances of a class. Their values can only be modified by methods or procs defined in the class definition.

[incr Tcl] can be used to combine any set of related procedures and data to provide a more organized set of code.

Control_Chart Class

A control chart class was implemented to combine the blt_graph widget, scale widget, and button widgets, along with procedures to manipulate and display the data, control limits, spec limits and warnings. The control_chart is basically a frame widget containing the various widgets. The manipulation of the widgets is accomplished through methods which access the various components of a control chart. The config method is used for setting the public variables.

4.4 Control Chart User Interface

There are two CAFE menu options required for the generation of a control chart, Design/Edit Chart and Display Chart. The Design/Edit Form interface for a control chart (as shown in Fig. 4-4) is divided into three sections: Graphical Display of the Chart, Method of Accessing Data, and Specifying the Amount of Data to be Displayed. The interface also contains a message window to clarify each field.

If the control chart is to be displayed the name must be specified. The name of the control chart must be unique, and no modification of the control chart is allowed if the chart is not owned by the user. If not owned by the user, the chart may be displayed but any fields modified will not be saved. The following is a description of the Design/Edit form.

Name The name given to the control chart and used to identify the control chart information which is stored in the Gestalt database.

Graphical Display of the Chart The Graphical Display of the Chart section includes the parameters which determine the basic design of the chart. A window containing no data will appear, and as the fields are changed the chart updates to reflect the changes to the control chart format. The changes are only saved when the Save button is pressed and only if the name of the control chart has been specified. The following is a description of the fields pertaining to the basic design of the chart.

Title The title to be displayed at the top of the chart.

Left Axis The label for the left axis of the control chart; this typically describes the process response being monitored by the chart.

Upper Spec The upper specification limit which will be displayed as an arrow on the left axis label at the specified limit. (Note: different from the upper control chart limit, 3σ above the average value of the data, which is calculated internally based on the measurements.)

Lower Spec The lower specification limit which will be displayed as an arrow on the left axis label at the specified limit. (Note: different from the lower control chart limit.)

Methods of Accessing Data There are three methods of accessing data: through the SPC identifier, the recipe, or through a file containing the desired data. The desired machine must be specified for the SPC id if there is more than one machine associated with a particular process. The machine name must also be specified if the recipe option is to be used. The machine name is not needed for the file option because the data will already be contained in the file. A description of the fields follows.

Machine The name of the machine being monitored.

SPC id The SPC identifier for the process/inspection step contained in the PFR link the process with the measurements.

Recipe The recipe of the process being monitored.

File The complete file name containing the data that will be displayed in the control chart.

Specifying the amount of data Two options are provided for specifying the amount of data to be displayed. This is for the initial generation of the chart and may be interactively changed once the chart has been generated. The first option is to specify the number of runs to be displayed prior to an end date. The second option entails specifying an initial date and a final date which will determine the amount of data to be displayed.

Number of Runs The number of runs being monitored prior to the end date.

Initial Date The initial date of the measurements being monitored.

End Date The end date of the measurements being monitored in the control chart.

4.4.1 Features in Control Chart

Horizontal Axis Shifter has the ability to gradually shift the horizontal axis view. The scale underneath the chart controls the horizontal view of the chart. The value of the scale indicates the leftmost run number shown in the control chart.

Zoom is controlled by the left mouse button. The region marked by the box created from dragging and releasing the left mouse button becomes the new view of the control chart. The middle mouse button restores the control chart to the original view.

Detail Information associated with a particular point is displayed in a new window when the right button is clicked on that point. The window contains information such as the date, time, lot name, actual measurements, etc.

Chapter 5

Accessing Data

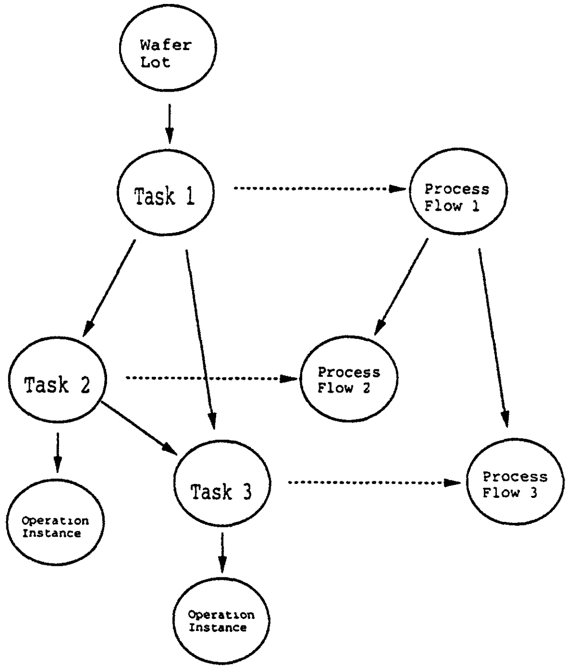
Accessing the appropriate data is a significant component to generating SPC charts. This chapter describes the process of obtaining data from the database using the PFR to link the process being monitored and the corresponding inspection process.

5.1 PFR

The Process Flow Representation (PFR) models wafer fabrication processes [11]. The model assumes a process is an operation performed on a product or a set of products. The operations consist of a sequence of suboperations. Associated attributes provide more detailed information about the operations; time-required, documentation, treatment, machine, and settings are examples of the attributes associated with an operation. Fig. 5-3 is an example of the textual PFR syntax. A programmatic interface to the PFR is used to access the data, needed for control charts as described in the next section.

5.2 Accessing the Data

The data used for the chart is obtained from Gestalt objects which store the readings of the various inspections. The `processflow` tree consists of database objects created from the textual representation of the PFR. The `task` tree structure is isomorphic to



Process Flow 2: Furnace Operation
Process Flow 3: Inspect Thickness

Figure 5-1: Task and Flow Tree Structure

the **processflow** tree structure. Each **task** tree is specific to a lot which undergoes the operations within the processflow. The task objects point to the **processflow** objects associated with them as well as operation instances associated with the **task**. The operation instance contains information such as settings, measurements, readings, etc. specific to the operation performed on a lot. The operation instance of a **processflow** which is an inspection contains the measurements needed for the control charts. Fig. 5-1 shows an example of both the **task** and **processflow** tree structures [11].

The readings of the output parameter are stored in data objects representing completed operation instances called **completedopinsts**. An operation such as an oxidation and the inspection associated with the oxidation are two separate processes and therefore are separate **processflow** objects. Fig. 5-2 shows the sequence of **tasks** and the **processflow** and **completedopinsts** associated with each task. Assuming Task 2 is an oxidation and Task 3 is an inspection of the oxidation, the data to be used for the control chart is contained in the completed operation instance associated

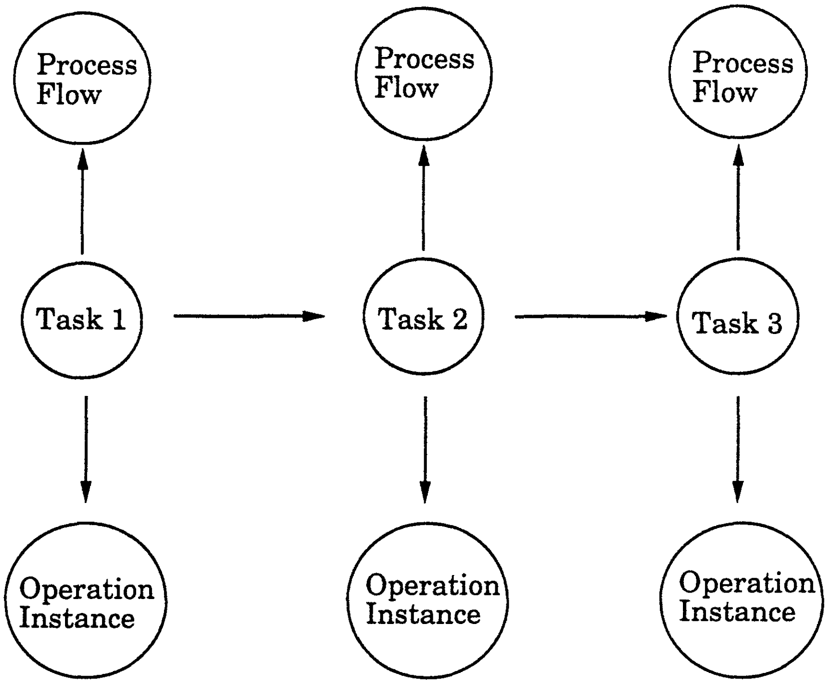


Figure 5-2: Task Sequence

with Task 3. The machine on which the oxidation was performed is contained in the operation instance associated with Task 2. The recipe (a specific set of settings for the machine in order to obtain a desired output) is also contained in the operation instance associated with Task 2. Prior to this work there was no link between a **processflow object** representing a process and the **processflow** object for the corresponding inspection. Therefore the problem of accessing the wrong data arises if measurements were not taken or not taken directly after the process of interest. The solution to accessing the correct data implemented here consists of the addition of SPC identifiers.

SPC identifiers link a process which performs some operation and the inspection process which provides the measurements of that specific process. The SPC identifier is a keyword string added to the advice field of both the process to be measured as well as the inspection step containing measurements for that process. Fig. 5-3 shows an example PFR containing the link between the furnace operation and the inspection-thickness. The keyword **:spc_process** and associated value added to the

```
(define dsro430
 (flow
  (:doc "Stress Relief Oxide")
  (:version
  (:modified :number 1.0 :by "Greg Fischer" :date "February 22, 1994"))
  (:advice (:opset "dsro430" :name "STRESS-RELIEF-OXIDE"))
    (:time-required (:hours 7 :minutes 15))
      (:body
       (operation
            (:advice (:name "RCA-CLEAN"))
            (:time-required (:hours 2))
            (:machine "rca")
            (:instructions " **RCA CLEAN IS REQUIRED** ..."))
       (operation
            (:advice (:name "FURNACE-OPERATION" :spc_process "dsro"))
            (:change-wafer-state
                 (:oxidation
                        :thickness (:angstroms (:mean 430 :range 20))))
            (:treatment
                 (furnace-rampup-treatment :final-temperature 950)
                 (furnace-dryox-treatment :temperature 950
               :time 6000)
                 (furnace-rampdown-treatment :start-temperature 950
                 :anneal-time (:minutes 30)))
            (:machine "tubeA1")
            (:settings :recipe 210)
            (:instructions "Add test wafer.")
            (:time-required (:hours 5)))
       (operation
            (:advice (:name "INSPECT-THICKNESS" :spc_inspect "dsro"))
            (:time-required (:minutes 15))
            (:machine "ellipsometer")
            (:instructions "Center Wafer oxide THICKNESS. Spec: 430 +/- 20A")
            (:settings :film-type "K2 (oxide)")
            (:readings :test-wafer-id-1 "Test wafer id-1"
                 :thickness "Center Wafer:top-1"
                        :thickness "Center Wafer:center-1"
                        :thickness "Center Wafer:left-1"
                        :thickness "Center Wafer:right-1"
                        :thickness "Center Wafer:bottom-1"
                  :test-wafer-id-2 "Test wafer id-2"
                        :thickness "Center Wafer:top-2"
                        :thickness "Center Wafer:center-2"
                        :thickness "Center Wafer:left-2"
```

Figure 5-3: Example PFR with SPC identifier

advice field of the furnace operation and the keyword :spc_inspect and equivalent value added to the advice field of the inspection-thickness link the process with the corresponding data.

5.3 Database Interface

The structure of the database is important in accessing the data. An object in the database consists of the name and attributes of that type as shown in Fig. 4-3. An overview of the application components needed to access the database and display the control chart is shown in Fig. 6-1. The search begins by finding all the completed operation instances which are created by the machine, for example the operation instance corresponding to Task 2. If Task 2 of Fig. 5-2 is the desired operation being monitored. The completedopinst associated with the task would be a member of the list. The task associated with each completedopinst is then determined, and the SPC identifier is obtained from the processflow object of the task. The next-leaf-task (an attribute of the task object containing the next task in a process) determines the following tasks. If the processflow of the next task contains the same SPC identifier for an inspect operation, the completedopinst for the inspect task (containing the post-process measurements) is accessed and the data is then saved in a string. If the processflow does not contain the same spc identifier, the processflow of the next task is searched until there is a match or there are no remaining tasks. The completedopinst for the inspect task (containing the post-process measurements) is accessed and the data is then saved in a string. The search begins again with the next completedopinst in the list. The measurements are saved in the cache object upon completion of accessing all of the desired data.

Chapter 6

Caching System

Due to the enormous amount of time that the search for the desired data will take each time the charts are generated, caching the data will speed up the time to produce subsequent charts. The information contained in the cache will be the data from the inspection operation, the time the data was generated, the machine the operation was performed on, the average and range of the data, as well as the lot name and the labuser who owns the lot. The SPC identifiers used to link the data with the process may also be used to access the cache. With a caching system in place the only data accesses needed are to update the cache with new data.

6.1 Caching

The caching of the data is accomplished through the use of a database object called `reportcache`. The object contains an attribute to store the timeinterval associated with the data. The timeinterval is the initial time of the first measurements and the final time of the most recent measurements stored in the object. From the CAFE menu the Display Chart option invokes the Common Lisp function, `trend-chart`, is invoked accessing or generating the `reportcache` object which contains the post-process measurements and additional information to graph the control charts. Fig. 6-1 shows a general overview of the progression of the application program. The control chart design/edit form requires a machine and the SPC identifier or the recipe in order to access the data to be charted. The information obtained by the form is stored in the database to be used when the control chart is generated. When the chart is in the process of being generated the cache object is accessed and, based on the timeinterval, the database is searched to determine if a new operation has been performed since the last update of the cache object. If a new operation has been performed the associated measurements are added to the **reportcache** object. The contents of the cache are written to a file in /usr/cafe/cache/trend under the name of the control chart specified, which is used by the Tcl/Tk script files to generate the control chart.

The cache is updated on each generation of the control chart. The frequency of the generation of the chart will determine the efficiency of the cache.

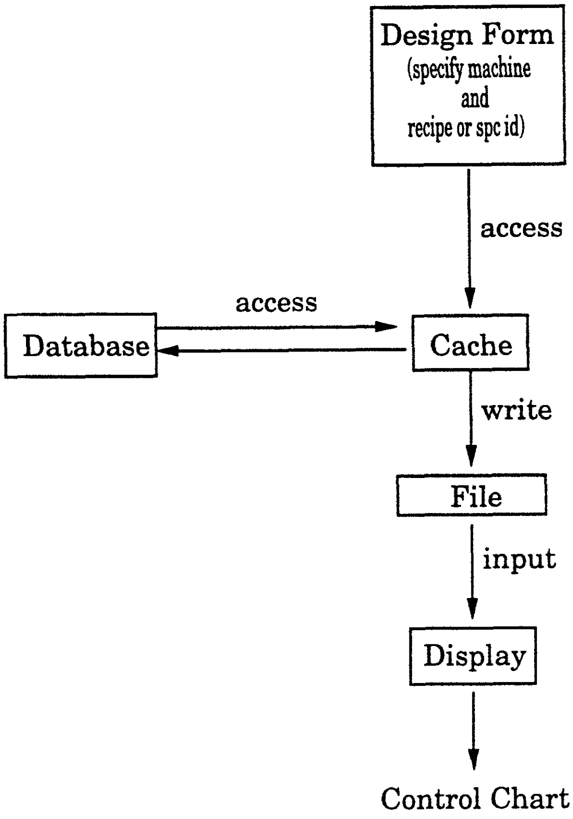


Figure 6-1: Overview of Program Progression

Chapter 7

Combining Control Charts

Control charts are primarily used to monitor one output variable of a process to determine if the process is in statistical control. The combination of charts to increase the detection of a process that is out of control and/or provide more information to aid in determining the assignable causes of variation investigated in this research.

Combining control charts of one machine producing the same output parameter at different target levels requires focusing on the average deviation of the output parameter from the average, rather than the average of the measurements. Assuming the standard deviation is approximately the same for both control charts, combining the control charts by using the deviations from the target instead of the actual measurements produces a standardized control chart with more information to help analyze if the processes are in statistical control and may help the operator in determining an assignable cause if the process is out of control. If the control chart indicates that the process is out of control this may be due to the machine or the other common factors involved in the process. Common factors to both processes include temperature, gas flow, possible raw material, the operator's error, etc. There are a huge number of assignable causes and combinations which may be the cause of a process being out of control and therefore narrowing the possibilities is always helpful to the operator. Determining more accurately the cause might entail looking at the processes individually. Combining charts of a machine with different processes may be extremely beneficial to small facilities where many different processes might be

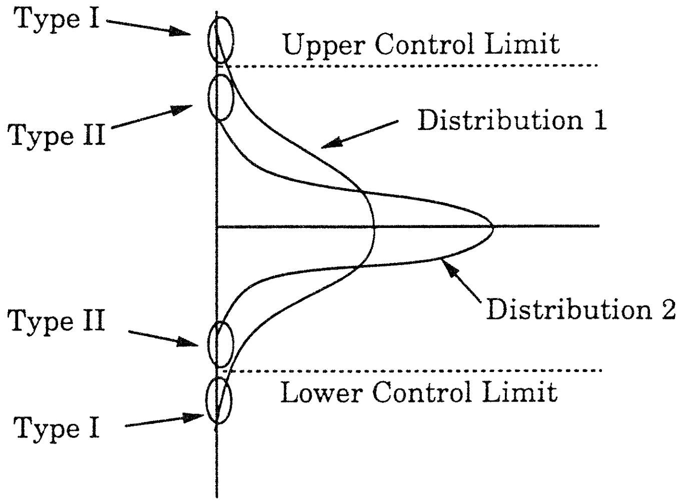


Figure 7-1: Type I and Type II errors

run on the same machine and therefore the individual control chart for each process would be produced from a small amount of data.

Combining control charts of two machines producing the same output parameter at the same target level is considered a group control chart [15]. The group control chart provides detection of the shifting output of one machine or both machines. The shift of only one machine implies an assignable cause associated with that particular machine, while a shifting of both machines implies an assignable cause which affects both machines. Shifts affecting one machine may be problems with the machine or the raw material or incoming wafers used for that process. Shifts affecting both machines may also be associated with the raw material as well as the operator and the environment interacting with the process. Group charts provide an overview of the performance of the facility on a particular process. It may be advantageous to combine charts of different machines with different target levels. A group chart based on the deviations of the output parameters may also be an option to analyzing two machines with different target levels, and may narrow the possible assignable causes of a process which is out of control.

The possibility of an inaccurate SPC analysis from the combination of control charts is primarily due to the assumption that the standard deviation of the processes are approximately the same. If the standard deviation of the processes are not the same the limits are no longer an accurate basis for SPC, and the SPC charts will produce both Type I and Type II errors leading to false warnings and no warnings when the process is in fact out of control. Fig. 7-1 shows a diagram of two distributions with different standard deviations and the location of measurements which might result in erroneous decisions because of the difference between the standard deviations. Assume distribution 1 is the wider distribution and distribution 2 is the narrower distribution. Type I errors occur from false warnings signalled by data points which fall in the portion of distribution 1 between the upper/lower control limits and the 3σ limits associated with distribution 1. The data points may in fact thus lie within 3σ of distribution 1's mean, but will falsely signal a control problem on the combined chart.

Type II errors are warnings which are mistakenly not signalled by the control chart. The Type II errors indicated on Fig. 7-1 correspond to data points in the tail of distribution 2 which are beyond the 3σ limit but which fall within the boundaries of the control limits.

Fig. 7-4 is a combination of two control charts; Fig. 7-3 (corresponding to distribution 2) and Fig. 7-2 (corresponding to distribution 1) are the individual control charts used to produce Fig. 7-4. Figs. 7-4, 7-3 and 7-2 show an example of the errors generated in the combination chart with standard deviations which differ by a significant amount. Run 13 on Fig. 7-3 is beyond the 3σ limit of the control chart signaling a warning that the process is out of control. The corresponding point in Fig. 7-4 is Run 26. In Fig. 7-4 a warning is not triggered by this data point producing a Type II error because the point falls within the control limits which have a wider range than the individual control chart. Fig. 7-4 at Run 17 has a warning indicating the process is out of control but looking at Fig. 7-2 Run 10 the point does not extend beyond the limit. This is an example of a Type I error, which signals a false warning.

The difference between Type I and Type II errors is serious. Although false warnings do not provide an accurate or efficient description of the process, a false warning is preferred to no warning. A Type II error will reduce the speed of detecting instabil-

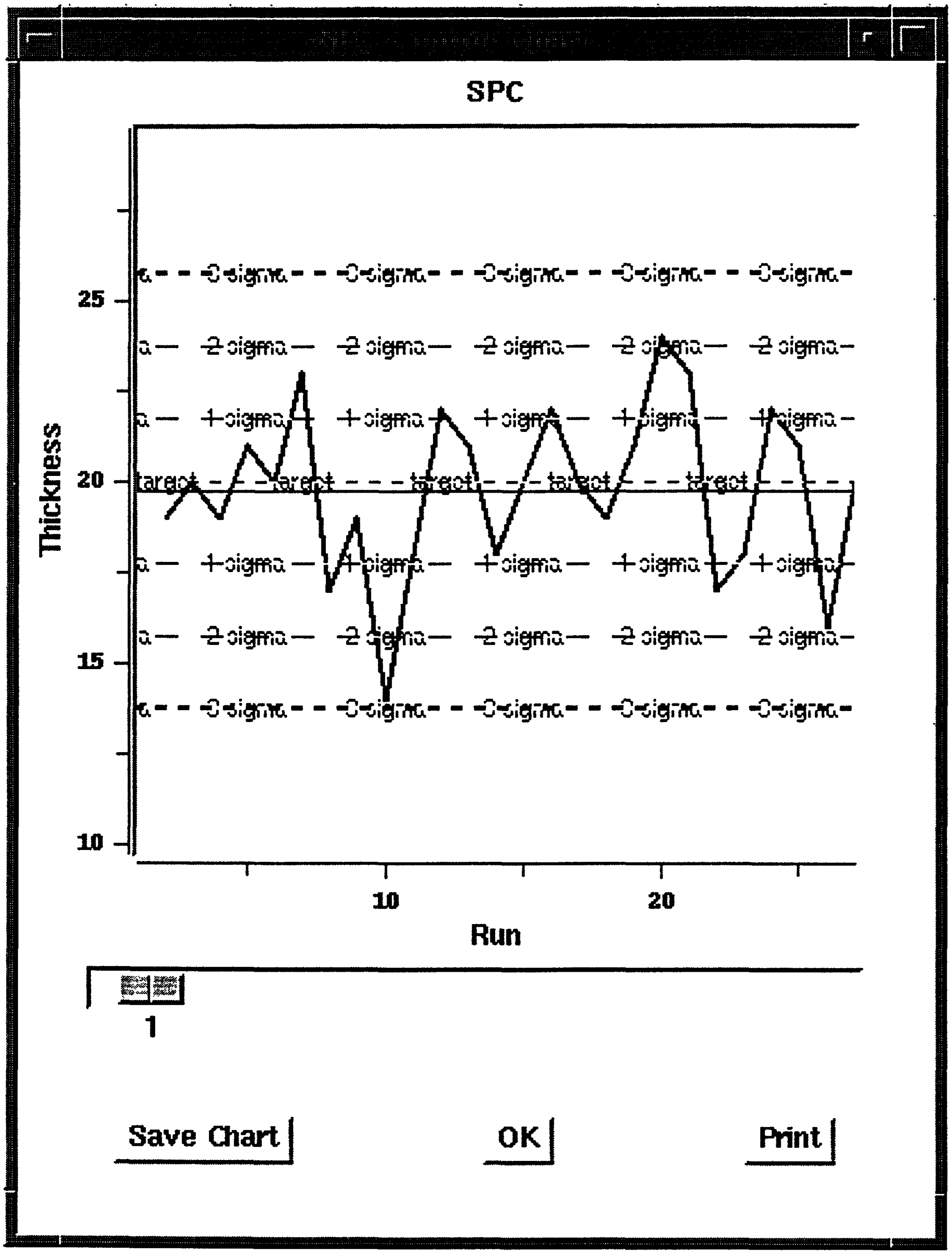


Figure 7-2: Example SPC chart of Distribution 1

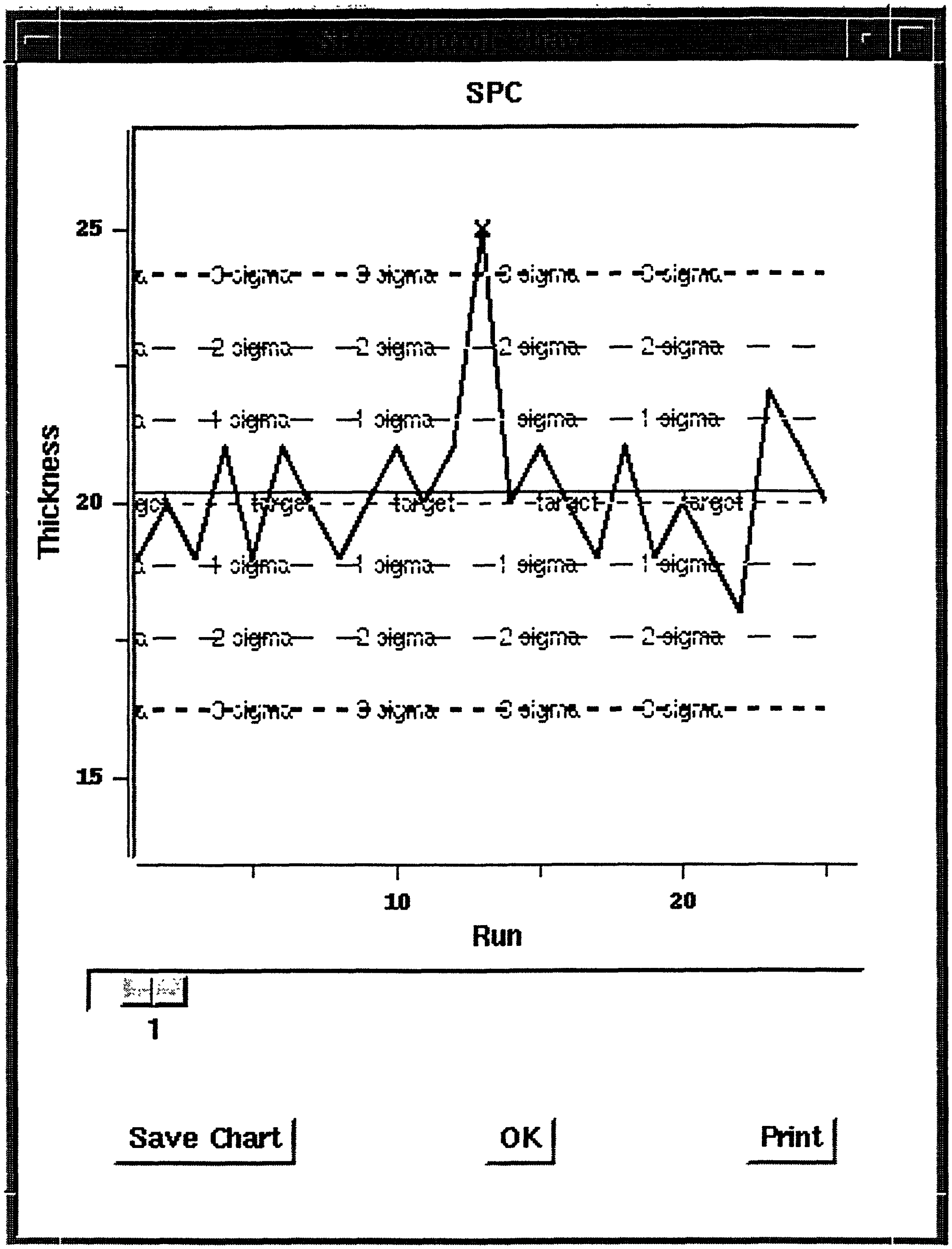


Figure 7-3: Example SPC chart of Distribution 2

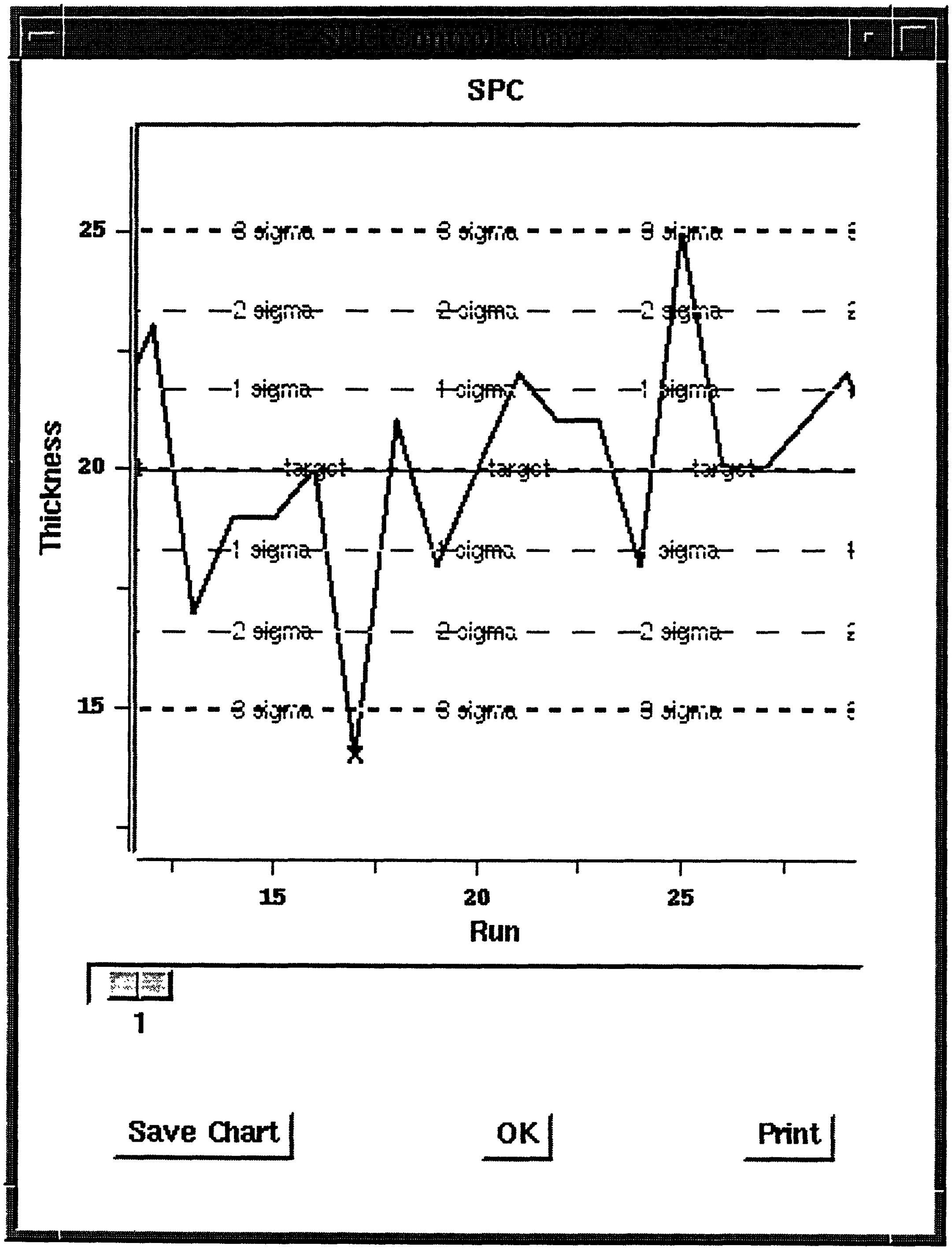


Figure 7-4: Example of Combination SPC chart

ities in the system. Therefore when combining control charts the standard deviation of the charts should be approximately the same to produce the most accurate control chart for the processes.

Combined control charts are beneficial in visualizing the entire performance of a manufacturing facility and providing insights into the assignable causes present. Combination control charts of different processes on the same machine provide information about the machine and other common factors between different processes. Specification limits are normally set based on consumer needs and engineering knowledge of the process capabilities. Looking at the control charts of processes producing the same output may be economically profitable by providing narrow specification limits of an output parameter based on the statistical history of all the processes, thus providing marketing with a more attractive set of specification limits implying a better quality product.

Chapter 8

Conclusion

The SPC charts for CAFE are beneficial to both operators and managers, providing a history of the performance of a particular process. Operators and managers are concerned about quality control. SPC charts help to achieve these goals by monitoring the variation of an output parameter and signalling warnings based on the stability of a process.

The individual components of the control charting application program are also useful utilities independent of their particular use in CAFE. The data gathered from the database and the advantages of the caching scheme can be utilized by post-process analysis programs; for example, run by run feedback control requires the same set of measurement data. Given any correct file format a control chart may be generated; the data need not necessarily be obtained from the database. The use of [incr Tcl] also provides the flexibility of using the control chart widget separate from the other graphical user interface components of the application. These features allow the separate components to be used independent of each other by other application programs in CAFE.

Improvements and enhancements can be made to provide the user with more choices for statistical process control of a process. The addition of the range control chart or the standard deviation control chart would aid in determining assignable causes affecting the uniformity of a process. There are many types of control charts available for monitoring processes and the implementation of these applications requires providing a graphical user interface and displaying the manipulated data using the control chart class and the data stored in the cache.

Currently processes which do not have an inspection process associated with them are ignored; it may be desirable to make note on the chart or the display the fact that a process has been performed at a particular time but no data is available for the control chart. The addition of the ability to display two or more charts on one chart may also be desirable and would require a minimal number of alterations. Implementing the improvements mentioned above based on the structure of the application program and the flexibility of Tcl/Tk should not be difficult and would increase the benefits and the functionality of the control charting application to operators.

Bibliography

[1] D. Boning. "A Tcl Interface to Gestalt". *CIDM Memo 93-4*, 1993.

[2] I.W. Burr. *Elementary Statistical Quality Control.* Marcel Dekker Inc, New York, N.Y., 1979.

[3] Western Electric Company. *Statistical Quality Control.* AT&T, Charlotte, NC, 1956.

[4] R.E. DeVor, T. Chang, and J.W. Sutherland. *Statisical Quality Design and Control.* Macmillan Publishing Company, 1992.

[5] B.S. Dhillon. *Quality Control Reliability, and Engineering Design.* Marcel Dekker Inc, New York, N.Y., 1985.

[6] A.J. Duncan. *Quality Control and Industrial Statistics.* Richard D. Irwin, Inc, Ill., 1974.

[7] A.V. Feigenbaum. *Total Quality Control.* McGraw-Hill, New York, N.Y., 1983.

[8] I. N. Gibra. "Recent Developments in Control Chart Techniques". *Journal of Quality Technology*, 7(4):183–192, October 1975.

[9] M.L. Heytens and R.S. Nikhil. "GESTALT: An Expressive Database Programming System". *ACM SIGMOD Record*, 1989.

[10] J.B. Keats and D.C. Montgomery. *Statistical Process Control in Manufacturing.* Marcel Dekker Inc., New York, NY, 1991.

[11] M.B. McIlrath, D. Troxel, M.L. Heytens, P. Penfield Jr. D. Boning, and R. Jayavant. "CAFE - The MIT Computer-Aided Fabrication Environment". *IEEE Transactions on Components, Hybrids, and Manufacturing Technology*, May 1992.

[12] M.J. McLennon. "[incr Tcl] - Object-Oriented Programming in Tcl". *AT&T Bell Laboratories*, 1993.

[13] D.C. Montgomery. *Introduction Statistical Quality Control.* John Wiley & Sons, Inc., New York, NY, 1991.

[14] J.K. Ousterhout. *Tcl and the Tk Toolkit (draft).* Addison-Wesley Publishing Company, Inc., 1993.

[15] M. Owen. *SPC and Continuous Improvement.* IFS Publications, New York, NY, 1989.

[16] E. Sachs, R. Guo, S. Ha, and A. Hu. "Process Control System for VLSI Fabrication". *IEEE Transactions on Semiconductor Manufacturing*, May 1991.

[17] E. Sachs, A. Hu, and A. Ingolfsson. "Run by Run Process Control: Combining SPC and Feedback Control". *Massachusetts Institute of Technology*, 1991.

[18] H.M. Wadsworth, K.S. Stephens, and A.B. Godfrey. *Modern Methods for Quality Control and Improvement.* John Wiley & Sons, Inc., New York, NY, 1986.