

SPC Software's Dirty Little Secret

By

**Stephen A. Wise
Director of Statistical Applications
Lyle-Kearsley Systems, Inc.**

**Copyright © 2000 Lyle-Kearsley Systems, Inc.
All Rights Reserved.**

No one likes to start over, especially after years of hard work. But that is exactly what SPC companies, that want to stay in business, are currently facing. Do these companies want you to know what I'm about to tell you? Of course not, they've spent years hiding the magnitude of their technical blunder.

The Root of the Problem

This problem is not readily visible and has been masked, until now. You see, it all hinges on how SPC software stores and retrieves data. To better understand the problem, let's go back to the days before computers – the days when SPC was performed with pencil and paper. Paper charts were (and still are) great for making real-time process adjustment decisions, however paper charts pose many limitations:

- Users have to manually calculate plot points and control limits.
- Completed paper charts have to be stored somewhere.
- Retrieving historical data from files of paper charts is time consuming.
- Generating summary reports takes hours, even days.

First Generation SPC Software Helped

Obviously, companies rejoiced when software became available that was capable of storing thousands of data points to a single file. Yes, computer files removed most of the apparent limitations of paper charting. The organization was simple; convert the paper charts into computer files – a file for each part. So what if there were lots of files, at least the paper was gone, no more hand calculations, and the data could be quickly charted and summarized. Each part file contained data for the part's monitored characteristics including associated specifications, control limits and traceability fields. Life was good for the SPC practitioners. Using computers, part files behaved very much like paper charts. However, given today's ever-increasing demands from shop floor data, what seemed to be a logical path back in the DOS days has now become a massive nightmare. SPC software developers that were first to market have certainly added features and improvements to their offerings, but the underlying data organization and setup logic still resembles the paper charts of old.

Databased SPC Software Missed Their Chance

With the advent of Microsoft Windows, relational databases and database servers, separate files were no longer necessary to house data from different parts. Some SPC companies have taken advantage of this capability and modified their software to write all data to a single centralized database on a shared server. This approach is far superior to managing individual files. Yet, when SPC software companies converted from files to a database, the file-based logic could have been abandoned and replaced with a far more powerful and flexible data organization. However, most SPC companies missed this major opportunity and went about writing code that stored data to the database using the same old part file logic.

Why was the Outmoded File-based Logic Preserved?

The answer is simple. SPC companies that converted their file-based products to databases had a problem – they needed a simple upgrade path for their existing DOS users – a path that offered a familiar logic and easy conversion of their legacy SPC files. The answer of many companies was to maintain the same file-based logic within their new database product(s). That is, part profiles (or groups) are defined in the top hierarchy of the database. All characteristics, specifications, control limits, tag fields, gage setups and so forth are configured within each part profile. Unfortunately, this part profile organization still contains the same limitations of separate files. Data from each part is stored to separate independent locations in the database. Since part data is independent, querying and analyzing data across different parts profiles behaves the same as if the data were still in separate files. Data from different part profiles must be displayed on different charts.

Does Your SPC Software have this Problem?

There are two ways to tell if your SPC software is plagued with this legacy problem.

1. Any software that uses multiple files to house measurement data from different parts or processes.

Files

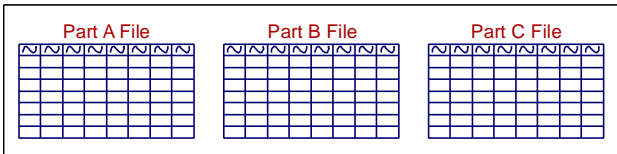


Fig. 1 Flat file SPC software. A separate file for each part.

2. Uses a database structure where the part (or process) is at the top of the hierarchy and test characteristics, specifications, control limits, tag fields and so forth are configured under part profiles (sometimes called part groups).

Groups

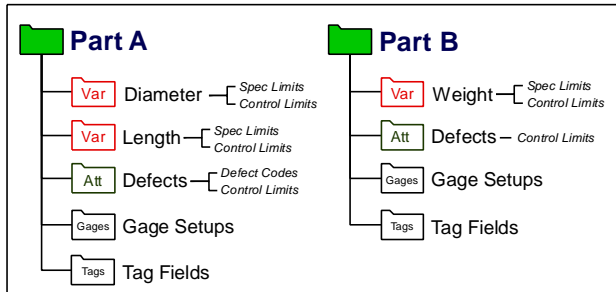


Fig. 2 Most databased SPC products still maintain the file-based logic. Each part group in the database is independent. Once stored to the database, data from different groups cannot be combined onto a single continuous time-ordered control chart. Data analysis of this type would require significant manipulations.

Both of these organizational conditions are identical in logic, the only difference is one approach uses multiple files and the other uses a single database.

SPC Thinking is Evolving and Users are Now Making Intelligent Requests

- I need to see the behavior of my lathe, across multiple part setups, all on the same control chart.
- I need to see, on a single box & whisker chart, which of my 8 turning machines produce the best outside diameters on aluminum parts, regardless of the part number. How about for hard steel parts? Now, only look at outside diameters between 1.5 and 2.5 inches regardless of the machine that ran the part.
- I need to do a Group control chart so I can plot all my production lines on a single control chart.
- I need to track salt, sugar and fat content all on a single control chart.

Many conventional SPC software providers claim they support the above functionality. And yes, sometimes with enough exporting, importing and manipulations, conventional-logic software can provide unconventional analysis. But the reality is; if you want to compare data from different parts or different processes, you need to know, prior to setting up the file structure, the types of comparative analysis you need. In addition, once the structure is setup, you typically cannot change your mind. Once data has been written to the file, very few changes can be made to the configuration. With these types of constraints, forget about having the flexibility in the software to keep up with your evolving thought processes, forget about producing process control charts that require data to be pulled from multiple

part files or across multiple part groups in a database and start thinking about shopping for your next SPC software system.

Making the Best of a Bad Situation

In an attempt to keep up with modern SPC thinking, some software companies added the ability to set up “short run” files (or short run groupings in the database) where multiple parts can be incorporated into one profile. These short run files can indeed provide true process control across multiple parts. But again, there is a catch: you need to know up front what to include in these short run files because once configured very few changes can be made. Also, if you have more than one machine that runs the same part, you will eventually have that same part in multiple short run process files. When it comes time to generate your customer Cpk reports, you better hope the software can go into each of those short run files and cherry-pick only specific parts and be able to combine them on another chart for the customer.

You see there is always a tradeoff when storing data using the conventional file-logic method. Your files must be either part specific or process specific. You cannot have it both ways. Given this tradeoff, most people will stick to the part-specific organization claiming that customer Cpk reports are more important than process control. No matter how many additional features SPC software may offer, if the data is written to the database using the file-based structure, the tradeoff between part control and process control must exist.

Somebody Got It Right – A Revolutionary Database Structure Offers Unprecedented Flexibility

Rather than part files, imagine a table within a normalized relational database filled with nothing but raw measurement values where each value (a row in the table) is related to items from other tables (parts, processes, test characteristics, tag fields, defect codes and so forth).

Under this data storage method there are no rigid relationships stored in the database. The part is no longer at the top of the hierarchy. Instead, there are three tables that share the top of the hierarchy. Those tables are:

- Part
- Process
- Test

We’ll call this the PPT structure. Given this method, every data value in the database is related to a part, a process and a test characteristic. Therefore, a user querying the database has full control over how to send data to the database and how to retrieve the data for display.

	Create Date	Part	Process	Test	Value
81	2000/01/17 00:54:28	A150-050-025	Lathe 167	OD - Loc A	1.502586
82	2000/01/17 00:54:28	A150-050-025	Lathe 167	OD - Loc A	1.501737
83	2000/01/17 00:54:28	A150-050-025	Lathe 167	OD - Loc A	1.504206
84	2000/01/12 14:22:11	3406	Engine Assembly	Final Inspection - Engines	1.000000
85	2000/01/12 14:22:11	3406	Engine Assembly	Final Inspection - Engines	1.000000
86	2000/01/12 14:22:11	3406	Engine Assembly	Final Inspection - Engines	2.000000
87	1999/12/17 11:12:04	12oz Big Life	Sultzon Can-O-Matic #5	Outer Lip Height	0.077500
88	1999/12/17 11:11:56	12oz Big Life	Sultzon Can-O-Matic #5	Outer Lip Height	0.076000
89	1999/12/17 11:11:47	12oz Big Life	Sultzon Can-O-Matic #5	Outer Lip Height	0.077000
90	1999/12/17 10:56:25	3406	Engine Assembly	Final Inspection - Engines	1.000000
91	1999/12/17 10:56:25	3406	Engine Assembly	Final Inspection - Engines	2.000000
92	1999/12/17 10:56:25	3406	Engine Assembly	Final Inspection - Engines	1.000000
93	1999/12/16 12:14:24	Loo-Hoo Can Lid	Sultzon Can-O-Matic #5	Outer Lip Height	0.077000
94	1999/12/16 12:14:24	Loo-Hoo Can Lid	Sultzon Can-O-Matic #5	Outer Lip Height	0.077000
95	1999/12/16 12:14:24	Loo-Hoo Can Lid	Sultzon Can-O-Matic #5	Outer Lip Height	0.076000
96	1999/12/16 12:13:47	Loo-Hoo Can Lid	Sultzon Can-O-Matic #5	Outer Lip Height	0.077500
97	1999/12/16 12:13:47	Loo-Hoo Can Lid	Sultzon Can-O-Matic #5	Outer Lip Height	0.077000
98	1999/12/16 12:13:47	Loo-Hoo Can Lid	Sultzon Can-O-Matic #5	Outer Lip Height	0.077500
99	1999/12/13 11:12:41	3406	Engine Assembly	Final Inspection - Engines	5.000000
100	1999/12/13 11:12:41	3406	Engine Assembly	Final Inspection - Engines	2.000000
101	1999/12/13 11:12:41	3406	Engine Assembly	Final Inspection - Engines	1.000000
102	1999/12/12 13:06:34	3406	Engine Assembly	Final Inspection - Engines	1.000000

Fig. 3 Database table containing all measurement values. Each value is identified with a Part, Process and Test. Organizing data in this manner allows for unlimited data manipulation and comparative analysis.

Benefits of a PPT Database Structure

Let's separate the benefits into three distinct categories:

- Setup
- Data entry
- Data display and analysis

Setup

Everything from part numbers, processes, test characteristics, specification limits, control limits, defect codes, assignable cause and corrective action codes, traceability fields and employee names is written to separate tables in the database. Everything! The only time combinations of these items get related is at the subgroup level. That is, only when data is entered into the database do these items get related, and only for that subgroup. No pre-conceived relationships need to be configured in this type of database structure.

Setting up a data entry configuration is independent of the items in the database. A data entry configuration can be designed to select from any part, process or test within the database. As a matter of fact, a data entry configuration can be designed to allow the user to re-select the part, process and/or test for each new subgroup. This is possible only because part, process and test relationships are not pre-defined anywhere. Only at the subgroup level do these items get related. A different data entry configuration for each part is no longer required.

Data Entry

From a shop floor data entry perspective, the user can click an add button and have the flexibility to select a different part, process and/or test for each entered data value. This is perfect for job shops that manufacture hundreds to thousands of different parts. No more searching through multiple lists of files to find the right one.

In addition to this flexibility, all the data display charts can be linked to the selections made during data entry. If, for example, a different part is selected during data entry, all the charts can dynamically change to reflect the current data entry items. In other words, if overall length from Part A was the last entered subgroup, the chart will be showing current and historical length measurements from Part A. If, however, the operator selects Part B during the next data entry and is now measuring the diameter, the chart that was showing length/Part A will dynamically switch to show the current plot point and historical data for diameter/Part B. This is possible because the charts' data selection (the properties that define what data is displayed) can be linked to the data entry configuration. If the operator makes a change to the part, process and/or test during data entry, the chart will automatically switch its data selection to match the most recent part/process/test data entry combination.

Theoretically, no matter how many part/process/test combinations there are, only a single data entry procedure is required. This is very different than the conventional logic where these part/process/test relationships must be pre-determined and are then unchangeable once data is saved to the file (or part group in the database).

Data Display and Analysis

This is where the most valuable benefits reside. Since charts are not tied to any pre-defined part file or configuration, they are open to display any conceivable part/process/test combination. One simply opens the chart's data selection and picks any combination of database items to include, or exclude, on the graph. The data available to be displayed to a chart is not at all determined by how data got into the

database. Therefore, multiple parts can be displayed on the same chart even though some of the part data was collected in real-time across multiple in-house workstations while other part data were imported from a supplier-provided text file and so forth.

One user generating a chart might wish to compare, all on the same control chart, how a particular machine creates an outside diameter regardless of the part number or feature size. This is no problem. At the same time, another user may want to create a single box & whisker plot displaying all the machines within the company that produce outside diameters and find the best one. Unlike the conventional hierarchy, the PPT database method provides the user with an endless number of query possibilities limited only by their imagination.

Does it Really Matter?

That depends – if your SPC data collection consists of monitoring processes dedicated to a single part and/or you are interested in tracking only outgoing part quality, the answer is no. The conventional data organization method is fine. In fact, under this scenario, any spreadsheet program is capable of producing adequate charts and reports. So if you fall into this category, why invest thousands in fancy SPC software?

On the other hand, if your SPC deployment requires you to monitor multiple processes used to produce different products, each having unique characteristics and specifications, different raw materials and different expected levels of variability, you need software based on the PPT logic. Until you make the change, you will never gain true process knowledge because any other software will have you locked into a file-based logic.

Did Anyone Get it Right?

Yes! Way back, in the late '80s, when the well-established SPC companies had their chance to make things right, a new SPC software company was just being formed. The original lines of code that employed the part/process/test logic have now evolved into what we know as *InfinityQS SPC*.

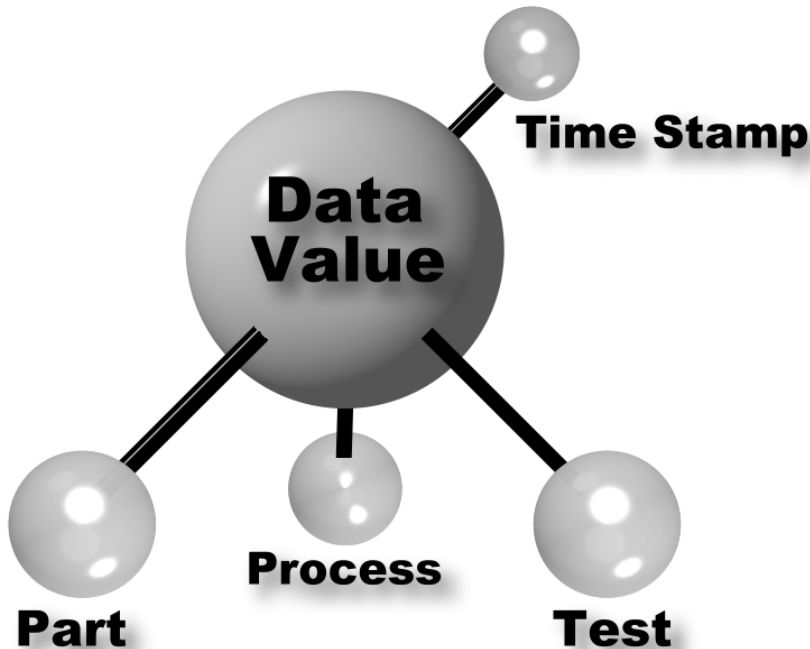
PPT Logic Traces Back to the Aerospace Industry

Did the founding developers of *InfinityQS SPC* know they were defining the next generation SPC software? Actually, no. The original designers of the code did not know they were writing something revolutionary. They did know, however, that from the onset, their aerospace environment demanded SPC software to be able to do incredible feats of shop floor data entry and charting manipulation. They needed to study parts, processes and tests both independently and in combination. The only way they knew to address these issues was to 1) start with a normalized relational database, 2) design a database structure where parts, processes and tests resided in independent tables and 3) create a data entry scenario where the part, process and test get related only to the measured value and only at the moment when the value is saved to the database. Under this structure, every subgroup in the database is tagged to a part, a process and a test. This is how *InfinityQS SPC* software achieves such great data manipulation and flexibility when creating charts and reports.

Try for Yourself

If you've ever wondered if there was true revolutionary SPC software on the market, you've found us. The choice is yours; you can keep waiting for your existing SPC software company to rewrite their code, or you can have the super-charged power of *InfinityQS SPC* on your desktop right now. There is absolutely no cost to give us a try. Just call 1-800-772-7978. Tell the operator that you are responding to the "Dirty Little Secret" article and you will receive a fully functional evaluation copy of *InfinityQS SPC*.

Say goodbye to SPC file based logic and say HELLO to PPT...



Stephen A. Wise is the Director of Statistical Applications at Lyle-Kearsley Systems. He holds a B.S. in Industrial Statistics. Prior to joining Lyle-Kearsley Systems, Mr. Wise worked as an industrial statistician at The Boeing Company, where he co-authored the groundbreaking D1-9000® specification, Advanced Quality System® for Boeing Suppliers. He is also co-author of the book *Innovative Control Charting*, published by ASQ Quality Press. Today, he helps companies on a daily basis implementing real-time production floor SPC and advanced statistical tools.

