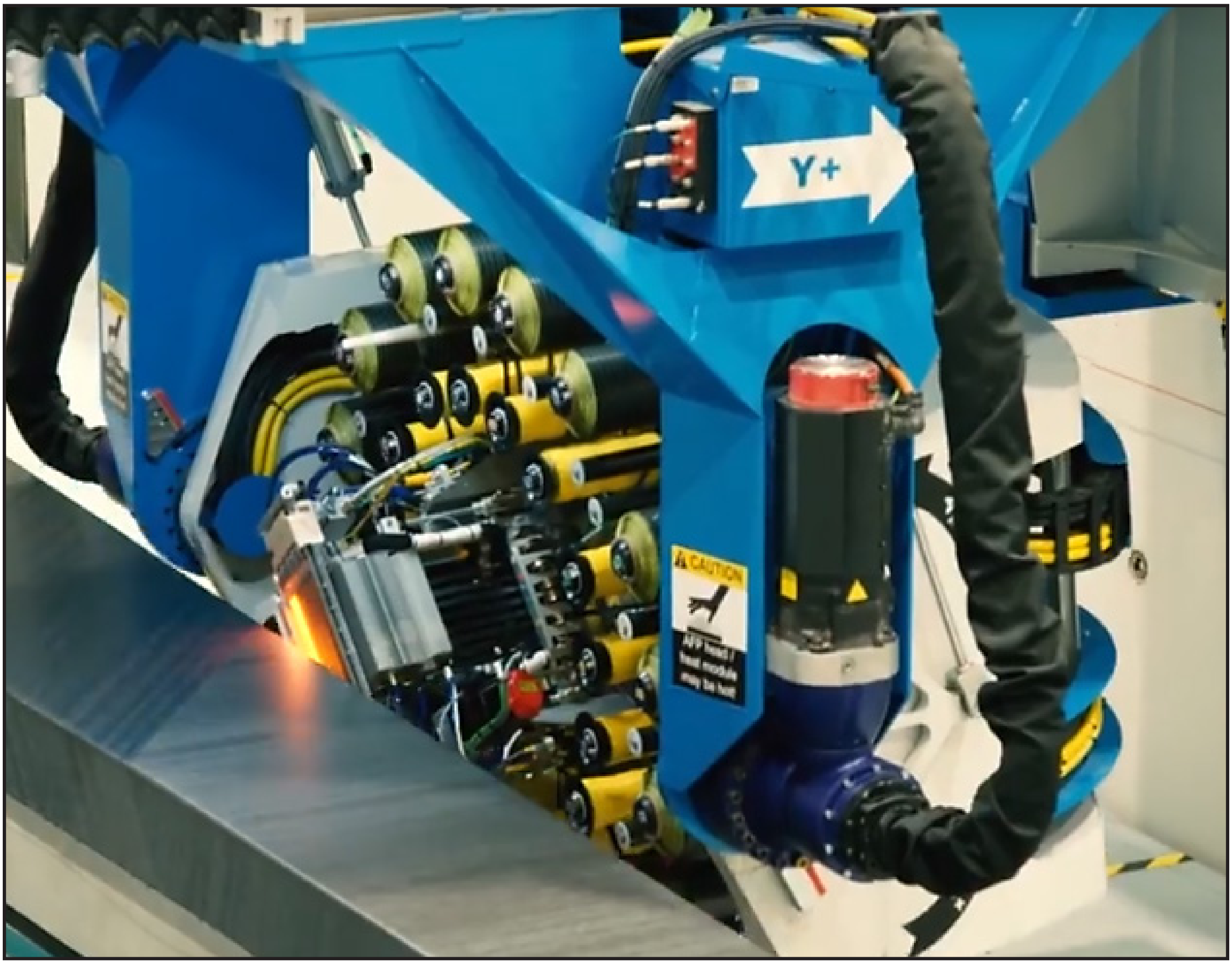


Software Configuration Management and Automated Secure Data Storage for Metrology Processes

by Robert Flynn and Schuyler Horky, Electroimpact

The mission of this project was to compensate and locate with six degrees of freedom (6DOF) several hundred two- or three-axis machines, all with an evolving software process and changing inputs. In this work, several significant data-handling problems arose: How were all input data, scripts, and other software versions to be kept under revision control with weekly or daily changes as well as several metrology teams running multiple shifts? How could engineers process the avalanche of incoming data without being overwhelmed by it? What could be done to prevent data loss due to operator error? How might all data be reliably archived and reports protected from accidental change? This article describes an effort to grapple with these questions and a gradual evolution from primitive to more sophisticated solutions that raised the bar a little on our own best practices for repeated metrology operations.

INTRODUCTION

At the outset of this work, the authors had just completed an extensive automated metrology project involving an aerospace wing assembly application with multiple automated workflows executed on 216 multi-axis CNC machines installed throughout eight fixtures. Each of these two- or three-axis machines was compensated to improve accuracy throughout its travel axes and then globally located within the coordinate frame for a given fixture. This experience allowed for the contemplation of software configuration management, automated data storage, and related issues, leading to the conclusions that: a) software configuration management is a necessity for automated metrology and that the "software" under management should include all data and other inputs which will affect the results; b) that

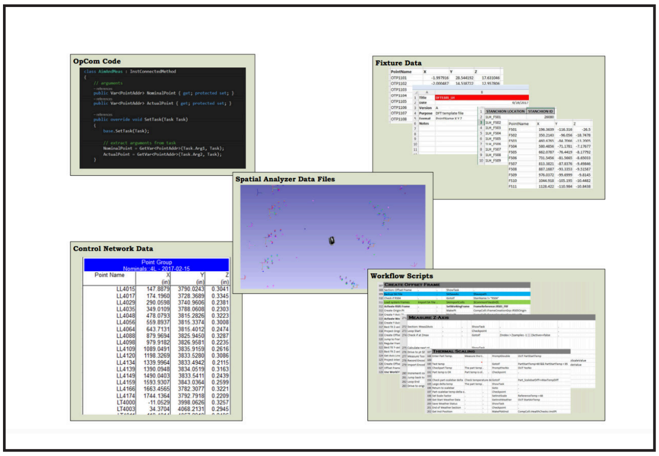


Figure 1. Some of data objects under configuration management

automated data storage and archiving is a best practice for automated metrology operations; and c) that automated data processing can be an effective tool to reduce engineering analysis workload. Before examining these conclusions, we must first consider what is meant by the term, "configuration management."

Configuration management of "software"

When thinking of software configuration management, we immediately think of software revision control and perhaps software that supports such control—and rightly so. But in the realm of automated metrology the definition of "software" under management must be extended to be comprehensive. This includes the workflow, by which we mean a program designed to automate a more generic program to execute a particular operation or set of tasks. An Excel macro is an example of a workflow. A measurement plan script generated out of SpatialAnalyzer of New River Kinematics (Williamsburg, VA) is another.

However, this definition also includes scripts which dictate the metrology process in addition to all inputs which affect that process. For example, automated metrology processes often exploit a control network, which is a set of carefully measured monuments or nests, usually scattered throughout the entire volume of the work area. Foundation reference systems (monuments glued into holes cored into concrete) and jig reference systems (nests permanently attached to tooling structures) are two examples of control networks. The dataset of the control network is a critical input, gets revised periodically, and must be revision-controlled. Other examples include template files for the measurement software (e.g., Spatial- Analyzer files) which may contain critical planes or other reference data. Previously measured tooling relationships and scalebar data may fall into the same category of critical data that requires revision control.

All of these elements of the automated metrology system must be correct for the system to operate properly. Thus, for a specific real-world automated metrology application, the "software" under configuration management is broadened to include these other inputs.

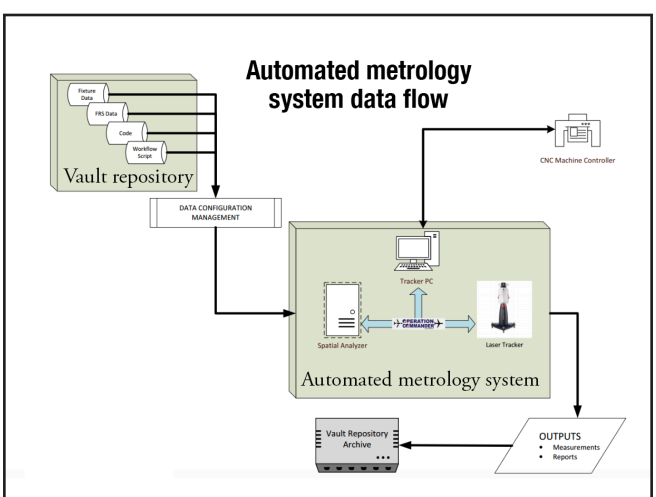


Figure 2. Automated metrology systems may depend on a wide assortment of inputs

Why revision control?

The necessity of revision control for such data is obvious. Control network data, for example, often changes only very slightly from one revision to the next, perhaps reflecting the changes incurred by curing concrete or other subtle effects. Changes of this nature are generally far too subtle to identify by a casual inspection of the numbers. Using such numbers in the process may make the numbers anywhere from a "little wrong" to "obviously wrong," or somewhere in between. Best practice demands that metrologists ensure that the correct numbers are used so that results are valid. Aside from all these considerations, the customer requires revision control. Having said all that, however, another practical question is raised: How can all these inputs be controlled in a user-friendly way?

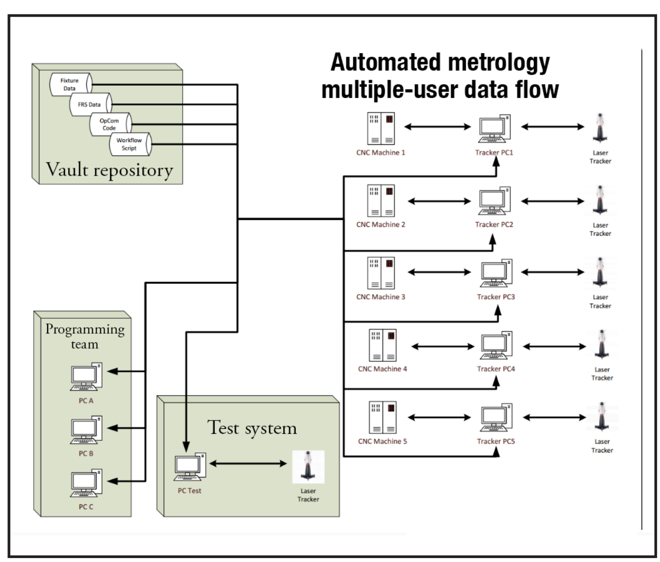


Figure 3. Increasing the number of users complicates configuration management and makes a "bulletproof" solution much more important to have

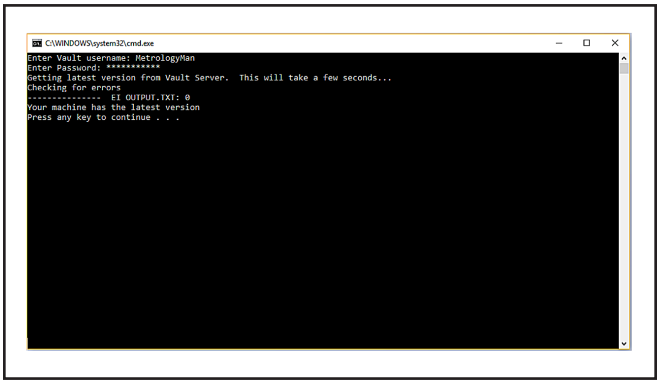


Figure 4. *A data-retrieval program can greatly simplify configuration management of complex software systems*

DATA-MANAGEMENT REQUIREMENTS

What are the requirements for a successful configuration-management scheme?

- It must accommodate implementation on multiple computers by multiple users.
- Because users are working some shifts which have no tech support, the solution must be "bulletproof" in execution.
- Any user must be capable of updating his or her own PC.
- Every user must be confident that the versions are all correct after updates.
- All necessary elements must be updated in the scheme; it must be comprehensive.

The solution to meet these requirements evolved. The team was already using an off-the-shelf revision-control software program, Vault from SourceGear (Champaign, IL) for revision control for all C# code, so it was natural to use Vault for revision control of the workflows as well. Critical data from coordinate measuring machines (CMM) or laser trackers were typically saved into a product data management (PDM) system for protection. Therefore, moving it over to Vault was a change, but one which greatly simplified the task of retrieval.

Initially, the team relied on training users to employ the native Vault tools for retrieval. We soon discovered that this needed more steps than we really liked and required more training. We also discovered that it was easy to forget a step and miss data. The solution was to automate the process. A software tool called "GetLatestOpCom" was created solely for the purpose of comprehensive data retrieval. Given an internet connection, this tool reaches into the Vault repository created for the project, allows entry of credentials, and retrieves all appropriate software elements required for the automated metrology system.

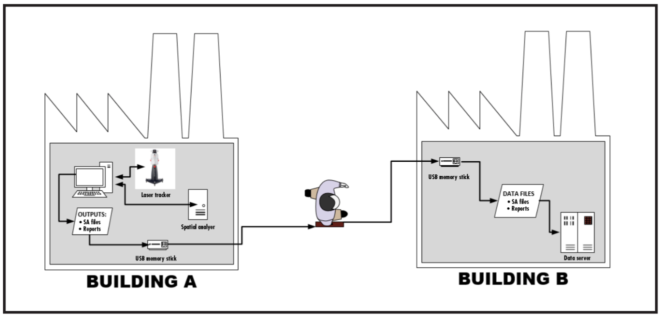


Figure 5. *Sneakernet… yes, it works but this method has its disadvantages*

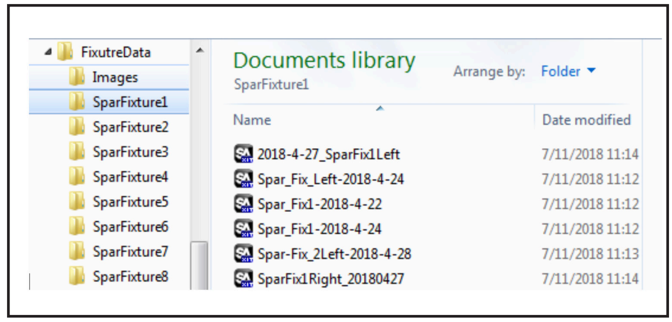


Figure 6. *Tired or careless users often use "almost the right file-naming scheme" with chaotic results, and the engineer analyzing the data is left to sort out and correct the file names—and to try to find the data needed for the next operation*

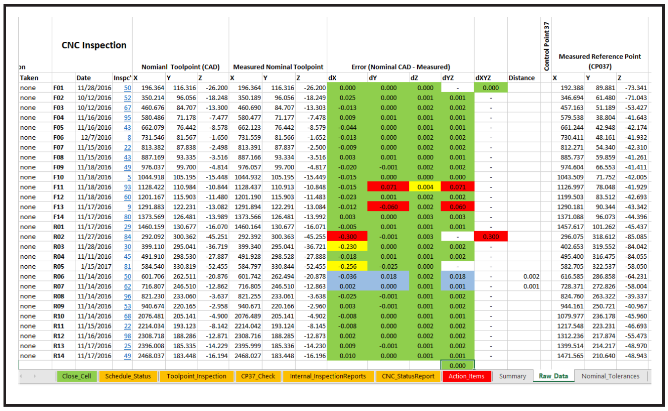
Now the training became greatly simplified and any user could easily update with assurance of success. These benefits led us to contemplate other uses for software automation and how it might apply to data storage.

AUTOMATED DATA ARCHIVE

Just about every laser tracker operator has likely used a handy thumb drive to transfer data from the tracker PC back to "headquarters" for permanent safe storage and analysis.

Probably every team that has used this "sneakernet" method for data transfer has run into a similar set of problems: missing thumb drives, occasionally forgetful operators, and fat-fingered users who copy the data to incorrect folders or misname the files or folders.

Therefore, when programming an automated metrology workflow it is very natural, indeed, obvious and inevitable, to carry the automation into the "wrap up" tasks of storing the measurements, reports, and any other outputs. But there are benefits to pushing this effort to also archive the data rather than simply store it locally on the tracker computer's hard drive.



on	CNC Inspection			Nomianl Toolpoint (CAD)			Measured Nominal Toolpoint			Error (Nominal CAD - Measured)						Control Point 37	Measured Reference Point (CP037)		
Taken		Date	Inspc'	X	Y	Z	X	Y	Z	dX	dY	dZ	dYZ	dXYZ	Distance		X	Y	Z
none	F01	11/28/2016	50	196.364	116.316	-26.200	196.364	116.316	-26.200	0.000	0.000	0.000	-	0.000			192.388	89.881	-73.341
none	F02	10/12/2016	52	350.214	96.056	-18.248	350.189	96.056	-18.249	0.025	0.000	0.001	0.001	-			346.694	61.480	-71.043
none	F03	10/12/2016	67	460.676	84.707	-13.300	460.690	84.707	-13.303	-0.013	0.000	0.002	0.002	-			457.163	51.189	-53.427
none	F04	11/16/2016	95	580.486	71.178	-7.477	580.477	71.177	-7.478	0.009	0.001	0.001	0.001	-			579.538	38.804	-41.643
none	F05	11/16/2016	43	662.079	76.442	-8.578	662.123	76.442	-8.579	-0.044	0.000	0.001	0.001	-			661.244	42.948	-42.174
none	F06	12/7/2016	8	731.546	81.567	-1.650	731.559	81.566	-1.652	-0.013	0.000	0.002	0.002	-			730.411	48.161	-41.932
none	F07	11/15/2016	22	813.382	87.838	-2.498	813.391	87.837	-2.500	-0.009	0.000	0.002	0.002	-			812.271	54.340	-42.310
none	F08	11/15/2016	43	887.169	93.335	-3.516	887.166	93.334	-3.516	0.003	0.001	0.000	0.001	-			885.737	59.859	-41.261
none	F09	11/18/2016	49	976.037	99.700	-4.814	976.057	99.700	-4.817	-0.020	-0.001	0.002	0.002	-			974.604	66.553	-41.411
none	F10	11/18/2016	5	1044.918	105.195	-15.448	1044.932	105.195	-15.449	-0.015	0.000	0.000	0.000	-			1043.509	71.752	-42.005
none	F11	11/18/2016	93	1128.422	110.984	-10.844	1128.437	110.913	-10.848	-0.015	0.071	0.004	0.071	-			1126.997	78.048	-41.929
none	F12	11/18/2016	60	1201.167	115.903	-11.480	1201.190	115.903	-11.483	-0.023	0.001	0.002	0.002	-			1199.503	83.512	-42.693
none	F13	11/17/2016	9	1291.883	122.231	-13.082	1291.894	122.291	-13.084	-0.012	-0.060	0.002	0.060	-			1290.181	90.344	-43.342
none	F14	11/17/2016	80	1373.569	126.481	-13.989	1373.566	126.481	-13.992	0.003	0.000	0.003	0.003	-			1371.088	96.073	-44.396
none	R01	11/17/2016	29	1460.159	130.677	-16.070	1460.164	130.677	-16.071	-0.005	0.001	0.001	0.001	-			1457.617	101.262	-45.437
none	R02	11/27/2016	84	292.092	300.362	-45.251	292.392	300.363	-45.255	-0.300	-0.001	0.003	-	0.300			296.075	318.612	-85.085
none	R03	11/28/2016	30	399.110	295.041	-36.719	399.340	295.041	-36.721	-0.230	0.000	0.002	0.002	-			402.653	319.552	-84.042
none	R04	11/11/2016	45	491.910	298.530	-27.887	491.928	298.528	-27.888	-0.018	0.001	0.001	0.002	-			495.400	316.475	-84.055
none	R05	1/15/2017	81	584.540	330.819	-52.455	584.797	330.844	-52.455	-0.256	-0.025	0.000	-	-			582.705	322.537	-58.050
none	R06	11/14/2016	50	601.706	262.511	-20.876	601.742	262.494	-20.878	-0.036	0.018	0.002	0.018	-	0.002		616.585	286.858	-64.231
none	R07	11/14/2016	62	716.807	246.510	-12.862	716.805	246.510	-12.863	0.002	0.000	0.001	0.001	-	0.001		728.371	272.826	-58.004
none	R08	11/14/2016	96	821.230	233.060	-3.637	821.255	233.061	-3.638	-0.025	-0.001	0.001	0.002	-			824.760	263.322	-39.337
none	R09	11/14/2016	53	940.674	220.165	-2.958	940.671	220.166	-2.960	0.003	-0.001	0.001	0.002	-			944.161	250.721	-40.967
none	R10	11/14/2016	68	2076.481	205.141	-4.900	2076.489	205.141	-4.902	-0.008	0.000	0.001	0.001	-			1079.977	236.178	-45.960
none	R11	11/15/2016	22	2214.034	193.123	-8.142	2214.042	193.124	-8.145	-0.008	0.000	0.002	0.002	-			1217.548	223.231	-46.693
none	R12	11/16/2016	98	2308.718	188.286	-12.871	2308.716	188.285	-12.873	0.002	0.000	0.002	0.002	-			1312.236	217.874	-55.473
none	R13	11/17/2016	25	2396.008	185.335	-14.229	2395.999	185.336	-14.230	0.009	-0.001	0.001	0.002	-			1399.514	214.217	-48.970
none	R14	11/17/2016	49	2468.037	183.448	-16.194	2468.027	183.448	-16.196	0.010	0.000	0.001	0.001	-			1471.565	210.640	-48.943
													0.000						

Figure 7. *When working with a large, complex system such as a wing assembly jig, a data-summary spreadsheet may become essential for planning*

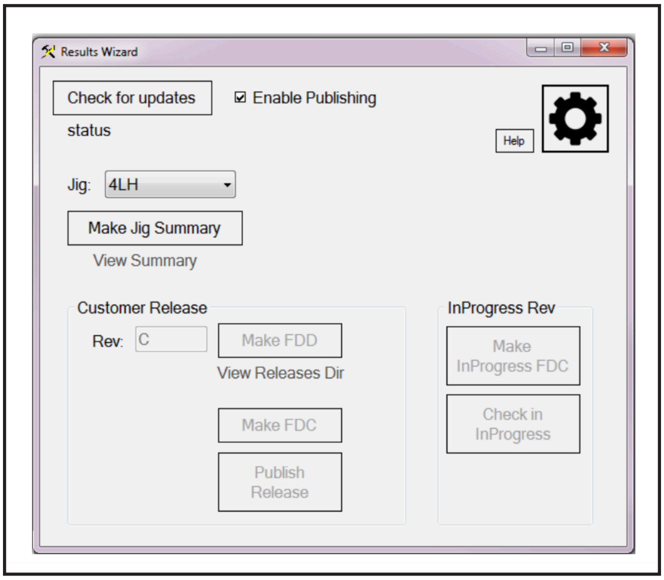
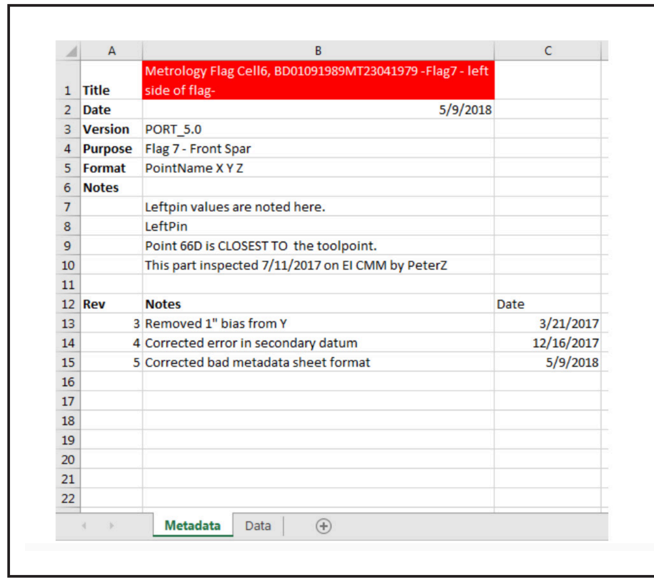


Figure 8. *The "Results Wizard" software tool became an important aid in reducing the workload of data consolidation*



	A	B	C
1	Title	Metrology Flag Cell6, BD01091989MT23041979 -Flag7 - left side of flag-	
2	Date	5/9/2018	
3	Version	PORT_5.0	
4	Purpose	Flag 7 - Front Spar	
5	Format	PointName X Y Z	
6	Notes		
7		Leftpin values are noted here.	
8		LeftPin	
9		Point 66D is CLOSEST TO the toolpoint.	
10		This part inspected 7/11/2017 on EI CMM by PeterZ	
11			
12	Rev	Notes	Date
13	3	Removed 1" bias from Y	3/21/2017
14	4	Corrected error in secondary datum	12/16/2017
15	5	Corrected bad metadata sheet format	5/9/2018
16			
17			
18			
19			
20			
21			
22			

Figure 9. *Using a file format that is friendly to metadata use has many advantages*

PROTECTING DATA

It is typically a requirement that measurement data be protected from tampering or accidental change and preserved against data loss. In some industries, such as aerospace, there are regulatory requirements to meet which further increase the value of maintaining data integrity. Therefore, a data archival or protection scheme becomes necessary.

The demands for automated storage, elimination of incorrect file-naming risks, and data protection combine to make a good argument for a formal software solution. Because the project team was using a software solution on the front end for revision control, it was natural to attempt an implementation of this on the back end as well. The team committed to the use of a wireless hotspot device to guarantee that an internet connection would always be available (the internet solution provided by the customer was not reliable). A method call was added to OpCom which enabled automated access to a SourceGear Vault repository dedicated to project data storage. During runs of the workflows the relevant data is automatically stored in the repository. Because the process is automated it is never forgotten. All file names used are programmatically generated, guaranteeing a total consistency in the naming scheme. Thumb drives and fat fingers are eliminated. The data stored is revision-controlled, and all versions remain on file so the data in its most raw form can always be reviewed if necessary.

The automatic archival effort turned out to be a clear winner. In addition to addressing all the major concerns, the rigid naming scheme enabled further automation taking place to simplify data analysis.

AUTOMATING ANALYSIS PREPARATION—RESULTS WIZARD

After the data is successfully retrieved and archived, the analysis task remains. To accomplish this, it becomes very useful to summarize the data across a jig to help visualize the next required task.

Initially, we tried to collate the data by creating a master Excel document which pulled data for individual cells from many individual spreadsheets. This posed two problems: First, in cases where an operation was run twice, the user had to manually select the most recent run and eliminate the older one. Second, Excel slowed to a crawl when drawing cell data from multiple documents. These problems were resolved by creating yet another custom software tool, the “Results Wizard,” which is run on a local computer by the engineer conducting the analysis. It retrieves the required data from the Vault, selecting only the latest “run” where there are duplicate runs for the identical tool. It then concatenates all the relevant data and copies it into a single spreadsheet, providing the required summary document.

The Results Wizard solution for data consolidation turned out to be an easy-to-use and reliable tool for daily monitoring of progress throughout the jigs, and it saved the team a great deal of time. Of course, every job would require a different, custom Results Wizard tool—it is not a universal solution. The point is that for certain tasks which require the consolidation of a great deal of data, the creation of a similar tool is worth considering.

LESSONS LEARNED

A few years of experience of wrestling with data taught some lessons that really stood out.

Use file formats which incorporate metadata

The use of a rigid file-control scheme such as that which is enabled by revision-control software offers great assurance that “all is well” with all the input data. This might encourage the thought that “it does not matter” what the

	y	z	dx	dy	dz	dyz	dxyz
196.366	-116.320	-26.511	-0.002	0.003	0.011	-	0.012
350.230	-96.055	-18.745	-0.015	-0.001	-0.003	0.003	-
460.739	-84.706	-13.202	-0.062	-0.001	0.001	0.001	-
580.532	-71.176	-7.174	-0.046	-0.002	-0.003	0.004	-
662.038	-76.441	-8.181	0.041	-0.001	0.004	0.004	-
731.589	-81.567	-8.657	-0.043	0.000	0.007	0.007	-

Figure 10. *Excel conditional formatting is invaluable in large spreadsheets as an aid to identifying out-of-tolerance numbers*

data structure is which is being saved. The question becomes whether one cares if point data comes in file formats *.txt, .xlsx,* or *.csv*. As it turns out, we do care. Best practice demands that a file format be used which allows easy revision verification. Although it is possible to do a file comparison (with the aid of software) to compare two *.txt* files to see if they are identical, this is hardly convenient. A far better practice is to use a file format which lends itself to the use of embedded revision data for the applications in question. Excel is a good example of such a format, because data may be kept on one sheet and revision information on another. The ability to quickly verify revision levels is invaluable, especially when debugging, and an embedded revision greatly facilities this.

Use exhaustive (diagnostic) reporting

One solid practice that came out of this project was that of creating an exhaustive diagnostic report strictly for the purposes of debugging. With an elaborate project involving complex interactions between CNC machines, trackers, and the tracker PC, there are many opportunities for errors. It soon became obvious that some reporting was required for debugging, which was not required by the customer. Initially such additional features were added to the standard report, but eventually this led to a bloated and potentially confusing report. Official (deliverable) reports should generally list only what the customer requests, therefore a separate diagnostic report was created which allowed unlimited expansion of reporting details.

Creating a diagnostic report is certainly an additional programming task, but when it comes time for debugging the rewards can be great time savings.

Use extensive checks to speed analysis and reveal errors

Complex programs offer more opportunities for errors than simple ones. When multiple programs are involved (in this example CNC code as well as workflow script) and multiple programming teams are contributing to the effort, the chance of errors increases even more. Meanwhile, identifying errors can often be difficult, especially when changes to numbers can be very subtle. Consider, for example, the effect of a very modest temperature change to scaling of a system—a programming error might apply the scaling in the opposite direction without being noticed.

Fortunately, nearly every number produced by a metrology system is subject to some form of reality check. In Excel sheets it is simple to set up nominal vs. actual checks for many measurements, verify known distances, etc. Almost every measurement and every calculation result typically falls within some very limited bound. Where possible, exact mathematical calculation should be used to cross-check program results. It can sometimes be tedious to put these into place, but extremely helpful in catching programming errors which otherwise might be opaque.

Once a respectable set of checks are in place they should be formatted appropriately. Flagging out-of-tolerance numbers in Excel using conditional formatting is very straightforward and provides a great tool for alerting the team to problems which might otherwise go unnoticed.

FUTURE EFFORTS

Although not groundbreaking, the practices of automated data retrieval and automated data storage were successful enough that we would like to incorporate them even into manual metrology jobs where appropriate. This involves many challenges since it is, by definition, a more open-ended, nonprogrammatic environment. But some effort is called for; metrology team leaders will certainly appreciate both having a simple and consistent method for distributing template files and other inputs to the team as well as data-handling schemes which will instantly and reliably store data into a protected and accessible location on a secure server.

CONCLUSION

As has been shown, automated metrology projects demand software configuration management for all program inputs in addition to automated data storage and archiving. Projects which require analysis of multiple measurements may also benefit from custom software tools to automate certain aspects of that data analysis. Implementing these systems and building these programs takes time but the rewards are great in terms of time saved and in the confidence of the whole team in the integrity of the data.