Maksi et al., J Inform Tech Softw Eng 2018, 8:4
DOI: 10.4172/2165-7866.1000246

Journal of
Information Technology & Software Engineering

Research Article Open Access

Deriving Common Factory Test Platform Requirements Using Historical Test Data

Maksi L, Berryman S, Brio A, Burkhardt A, Elder S, Ferkau S, Gharbiah H, Lynch K* and Risch Q

Engineering and Information Technology, Raytheon Corporation, Tucson, AZ, USA

Abstract

Businesses that produce many complex products inevitably have disparate test requirements. Over time test solutions diverge despite having similarities contributing to increases in test cost. Mergers and acquisitions further complicate having common, well-understood, low-cost test platforms. By comparing the population of parameters tested across all products insight can be gained as to what groupings of parameters could have shared, common test platforms. The method described in this paper, in use today, describes a data mining and statistical approach to identify groups of test capability by testing ranges and limits, leveraging decades of historical testing data across a wide range of commercial products to design common test equipment at a large aerospace manufacturing organization. Clustering methods were used and evaluated to determine common factory test platform requirements, and compare them against existing test platform vendor capabilities.

Keywords: Common test platform; Manufacturing test; Aerospace manufacturing, Manufacturing test data

Background

Aerospace manufacturing test organizations have long realized there are cost benefits of having common test architectures and common test platforms [1,2]. This was understood at Hughes Aircraft before it became Raytheon [3], and after [4]. Digital approaches to capturing, characterizing, and analyzing parametric test data for requirements can be found at the Army circa 1983 [5], however this effort was not successfully implemented. More recently requirements for a common avionics factory test platform were identified by Sonnenberg et al. [6]. All of this research suggests that there is a general acknowledgement that common test platforms save money in manufacturing; however, a concerted effort is required to consolidate test platforms to realize the savings. One attempt at consolidation, as a common factory test platform is described [7].

The founder of Hughes Aircraft, Howard Hughes, purchased most of the missile products they produced from several different firms. Raytheon acquired these and other products through mergers and acquisitions resulting in a portfolio of disparate systems with some many decades old. These products have been upgraded over the years increasing the complexity of test requirements. Avionics products with embedded systems such as these also have very long development times, making the recovery of original formal test requirements difficult [8]. The result is a portfolio of products that were designed and tested under differing philosophies and architectures. This is a challenging environment to design and develop a common test architecture, and provided a significant opportunity for the analysis described in this paper.

Introduction

Today one business unit at Raytheon supports over 700 distinct test sets, representing a diverse portfolio of complex systems. Over the years products were designed and acquired, unique test sets were also designed and acquired. System failures, obsolescence and the cost of supporting a vast number of unique test sets motivated this business unit to develop common test platforms that could support multiple diverse missile programs.

An initial effort attempted to manually document testing requirements for our complete set of products. The initial estimate of effort required to complete this task was over 40 person-years (at 4 weeks of subject matter expert time per test set). The most challenging aspect was that it would require a highly skilled engineering team that is in critically short supply. Such a pull of our talent would cause a huge strain on our need to deploy new products. Clearly, a method or process that would reduce the expected engineering workload would provide cost savings and a reduction in schedule risk.

The first step in our approach to design common test platforms was to build a set of test requirements that bound existing test measurements in every dimension. The original plan was to compile a set of requirements from an exhaustive search of Test Requirements Documents (TRDs) from over 100 of our current products. In the 1970's, MIL-STD-1519 specified the preparation of test requirements documents [9]. A test requirements document contains detailed specifications for every measurement that must be made to ensure the unit under test meets design requirements. A single test may be comprised of over 1000 test measurements. The standards associated with test platforms have changed considerably over the decades, with MIL-STD-1519 and its successor MIL-STD-1345B being retired, and the introduction of IEEE's ATML (Automatic Test Markup Language). This ensured consistency with test component standards IEEE 1636.1 (Test Results) and IEEE 1671.3 (UUT {unit under test} description) [10]. While automated test requirements documents have been considered as early as 1987 by Rockwell [11], implementing automation is made difficult in our environment where so many different products and test sets were inherited with existing documentation.

Many of the test requirements documents for our oldest products exist only in paper form. Deriving a complete list of test requirements from these documents would have required compilation, translation,

***Corresponding author:** Lynch K, Engineering and Information Technology, Raytheon Corporation, Tucson, AZ, USA, Tel: 0115207462050; E-mail: kevin_j_lynch@raytheon.com

Received October 01, 2018; **Accepted** October 25, 2018; **Published** October 31, 2018

Citation: Maksi L, Berryman S, Brio A, Burkhardt A, Elder S, et al. (2018) Deriving Common Factory Test Platform Requirements Using Historical Test Data. J Inform Tech Softw Eng 8: 246. doi: 10.4172/2165-7866.1000246

and expert review, each with their own difficulties. For example, we estimated that it would require a year and substantial expert review just to compile data needed to build the database for the test requirements analysis. The elapsed time meant that we would forego opportunities for commonality in the interim, so we searched for a different method to derive test requirements.

Approach

Instead of relying on an analysis of test requirement documentation we devised a plan to reverse engineer test requirements from the complete set of historical test data for all products into a common database. The process we used, described in this paper, is depicted in Figure 1.

We believed we could quickly evaluate what the overall test requirements should be based upon parametric values and units of measure in the existing test data. The parameter values would establish the ranges exercised in the test equipment. The units of measure would determine what testing capabilities are needed. The test data typically included the upper and lower limits of all test results. These limits were analyzed to determine the accuracy of measurements needed based upon the company standard Test Accuracy Ratio (TAR), the ratio of the upper and lower limits of the testing parameter, divided by the accuracy of the measurement. This analysis is crucial because this ratio is directly related to the confidence in the measurement. While the industry standard for metrology labs for test accuracy ratio is 4:1, this business unit's goal was 10:1, an earlier and higher historical standard [12].

The test requirements had to be developed to adequately test components from multiple programs. The goal was to identify specific test requirements for a common test platform. The current authors' primary requirement for a common factory test platform was to have maximum test coverage with minimal features, to minimize cost. To accomplish this, we compiled all of the historical testing data into a single database, and then analyzed the historical test parameter data across measurement unit types (e.g., ohms, amps, degrees Celsius, seconds, volts, MHz), as found useful [13].

Prior to performing any statistical approach it is necessary to pull the historical test data, group it into the corresponding measurement unit type (e.g., ohms, amps, degrees Celsius, seconds, volts), and apply the appropriate conversion formula to convert the data to a common measurement unit for each grouping type. The code has been automated to pull a list of units from a data warehouse, group the measurement unit types, and apply the appropriate conversion formulas. This necessitated harmonizing different historical measurement unit names, and dealing with missing data. One sided missing spec limits, instances where the lower spec limit (LSL) is less than the natural lower spec limit (e.g., LSL is -10 Ω, but the natural lower limit for Ohms is 0) or the upper spec limit (USL) is greater than the natural upper spec limit, are replaced with the natural lower spec limit and natural upper spec limit respectively from the stored data file. Clustering by units of measure helps designers understand how to optimize the coverage of a particular test platform. Many of the spec limits for current test measurements fell into a small number of clusters. We could match the test requirements to the range of testing from each vendor's equipment. We initially used a K-means clustering algorithm to determine the clusters [14], representing potential common test platform configurations.

In the database we developed our units of measure analysis, the upper vs. lower limits of testing were plotted on a single graph (Figure 2 shows a high-level example for voltage measurements). This allowed for quick inspection and rudimentary analysis. The clusters were signified by different colors, and provided a quick visual way to examine results that could recommend test set requirements.

We added a count of how many units of measure fall into or out of the testing capabilities contained in the clusters we created (Figure 3). This enabled us to use the amended data in R's Shiny data analysis tool. By completing this analysis we were also able

to visualize how many of the test measurements conducted by the 700+ test sets would fall outside the limits of each testing range cluster.

The initial K-means clustering approach used in our analysis was expanded to include multiple clustering algorithms to suggest

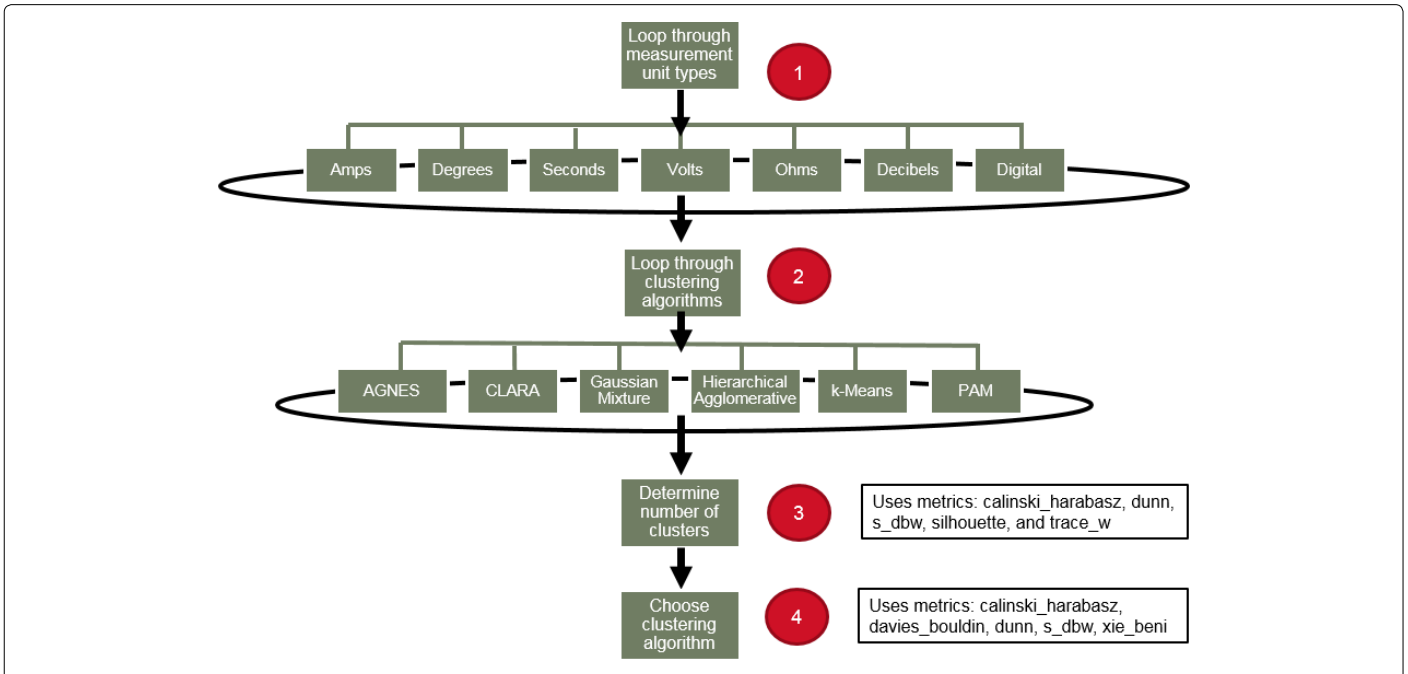

Figure 1: Process to identify clusters for common test platform configurations.

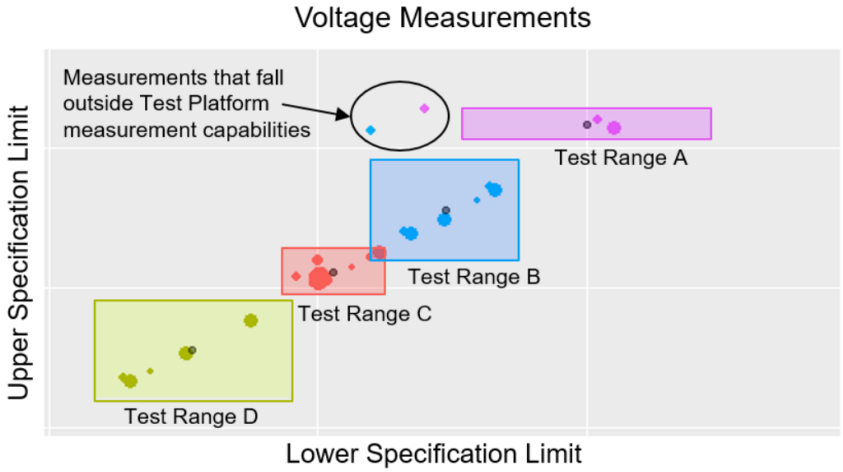

Figure 2: Test requirements vs. test platform capabilities.

LSL	USL	Cluster	Frequency	Within CFTP
-120	-107	1	2502	T
-120	-90	1	3	T
-120	-35	1	5	T
-118	-107	1	148	T
-116	-107	1	2200	T
-100	100	3	2	F
-58	-35	3	50	T
-56	-30	3	200	T
-55	-25	3	172	T
-35	8	3	27388	T

Showing 1 to 10 of 58 entries Show 10 entries

Figure 3: Data by units of measure.

common factory test platform configurations [15]. Different clustering algorithms will perform better than others depending on the underlying data set. Clustering algorithms used were: Hierarchical Agglomerative, AGNES, PAM, CLARA, K-means, and Gaussian Mixture Model. Where applicable distance matrices are calculated using Euclidean method. The combination of algorithms performed for each measurement unit type was limited by the computing power available. The algorithms were coded in R and skipped a particular clustering algorithm if it was deemed not enough memory was available. For each of the algorithms ran, multiple internal criteria were used to determine the appropriate number of clusters. The internal validation criteria were used to measure compactness or cluster cohesion, separation, and connectivity. These criteria include Calinski-Harabasz, S-Dbw, Trace-W (within sum squares), Dunn, and Silhouette. An example output is in Table 1.

Depending on the available computing power, not all methods were used. The algorithm chose the maximum criteria given the available capacity. The number of chosen clusters was returned for each internal criteria run and ultimately determined by the maximum agreement among the criteria. The above mentioned process steps are performed for every measurement unit type for each clustering algorithm, except the Gaussian Mixture Model. The Gaussian Mixture Model applies a maximum likelihood estimation and Bayes criteria to identify the most likely model and number of clusters. It selects the optimal model based on Bayesian Information Criterion (BIC) for Expectation Maximization (EM).

Subsequently, the clustering algorithms performed with the chosen number of clusters were compared to one another using internal validation criteria. The criteria indices used to compare the algorithms were Calinski-Harabasz, S-Dbw, Xie-Beni, Dunn, and Davies-Bouldin (Table 2).

Where the measurement unit type's data volume exceeded the threshold a stratified sample was taken for each cluster. The internal criteria were calculated on the sample for each algorithm ran and compared to one another. Where feasible (below the threshold) the entire data set was used to calculate the internal criteria indices for each algorithm. The chosen clustering algorithm was determined by the maximum agreement among the criteria indices. In the event of a tie S_dbw was used as the deciding factor. The intent was to choose the clustering algorithm that performs best as measured by compactness, separation, and connectivity for each unit of measure. Depending on the underlying data set different algorithms performed better than others for that unit of measure.

The results provide quantitative and visual support, enabling

Indices		Number of clusters								
Index	Rule	2	3	4	5	6	7	8	9	10
calinski_harabasz	Max	6.55E+04	2.52E+05	4.34E+05	3.56E+05	9.43E+05	1.30E+06	1.21E+06	1.22E+06	1.12E+06
s_dbw	Min	2.04E+00	9.91E-02	1.57E-01	3.07E-01	1.32E-01	9.36E-02	6.59E-02	4.04E-02	3.92E-02
trace_w	Max diff	1.18E+08	3.58E+07	1.58E+07	1.45E+07	4.66E+06	2.84E+06	2.62E+06	2.28E+06	2.20E+06
dunn	max	6.80E-02	7.94E-02	2.15E-02	1.52E-02	1.52E-02	1.52E-02	1.52E-02	1.52E-02	1.52E-02
silhouette	max	5.59E-01	8.54E-01	8.00E-01	8.21E-01	8.33E-01	8.19E-01	8.02E-01	7.86E-01	7.79E-01

Table 1: The indices in the above table are calculated to determine the optimum number of clusters for the clustering algorithm applied. A similar table was developed for each clustering algorithm.

Indices		Clustering algorithm		
Index	**Rule**	**Gaussian Mixture**	**CLARA**	**K-means**
s_dbw	Min	3.04E-01	9.91E-02	1.87E-01
calinski_harabasz	Max	6.61E+05	2.52E+05	2.82E+05
xie_beni	Min	5.75E+01	1.43E+00	7.45E+00
dunn	Max	5.08E-03	7.94E-02	3.40E-02
davies_bouldin	Min	4.27E-01	2.82E-01	4.70E-01

Table 2: The indices in the above table were calculated to determine the optimum clustering algorithm for the unit of measure being evaluated. A similar table was developed for each unit of measure.

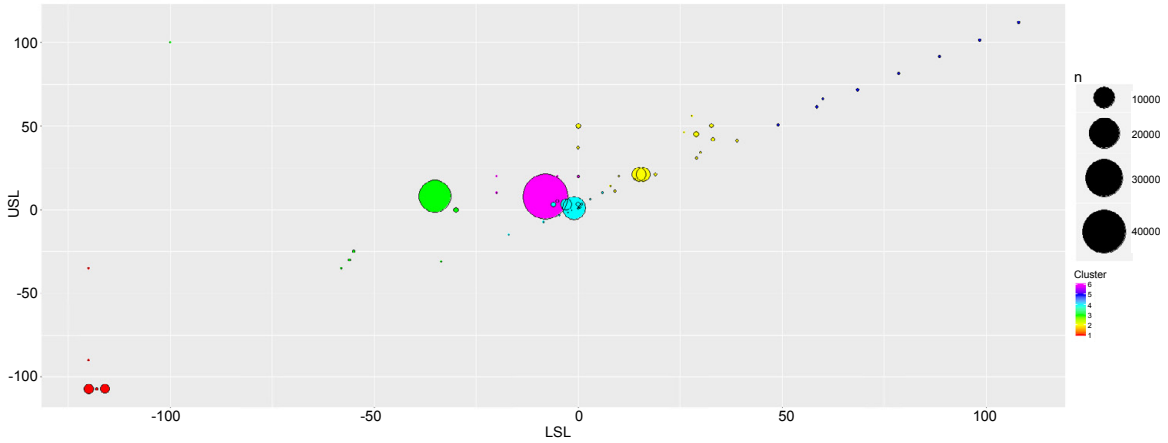

Figure 4: Example output of tests, frequencies, clusters.

common factory test platform designers to map products to test equipment coverage across factories, and facilitate migration of existing factory configurations to common test platforms. The chosen algorithm results were stored for each measurement unit type and were displayed via a basic web application utilizing RStudio's shiny. Figure 4 is an example output that displays the lower and upper spec limits color coded by cluster assignment and sized on frequency of occurrence.

Figure 5 plots a two dimensional clustering plot based on the chosen clustering algorithm partition. Component distances and shading are included. The clustering plots helped designers understand the implications of choosing a particular clustering algorithm.

For every cluster in each clustering algorithm, such as Figure 6 below were produced. Figure 6 lists all the unique lower and upper spec limits for the selected measurement unit with the assigned cluster and frequency of occurrence. These tables helped designers better understand the relationship between ranges of test measurements, their frequency, and specific clusters.

Figure 7 shows an example of mapping the same clustered data from Figure 4 onto a proposed design for a common test set. The capabilities of a test set are represented by the colored boxes. All data points falling outside a colored box would require an individual, non-common test set or a change to the test requirement that would meet the testing capability of the common test set.

The clustering analysis provide great insight as to what current test sets are able to test the products given the range of the historical test data; the next step was to analyze the existing capabilities of test sets and identify gaps between the testing requirements and the capabilities of the common test equipment. For instance, Raytheon had already purchased a test platform from an outside vendor intended to be used by multiple programs. We were tasked to determine if any additional test articles could be tested on this platform. To develop and test our method, we created a table of capabilities of the purchased system and used that data to represent a 'what if' design. The capabilities of this test platform were included in the clustering graph shown in Figure 6. We were able to compare the historical test data to a database of common test equipment capabilities. We conducted a simple greater than/less than analysis between the test data and the common test equipment capabilities, identifying each of the testing gaps between the test articles and the common test equipment. A count of units of measure was used to quantify the testing complexity. The number of gap adaptations required and degree of complexity are both related to cost and schedule. If the test article identified had a high percentage of units of measure that fit within the capabilities of a common test platform, we surmised it would be the easiest to deploy onto a common test platform. The

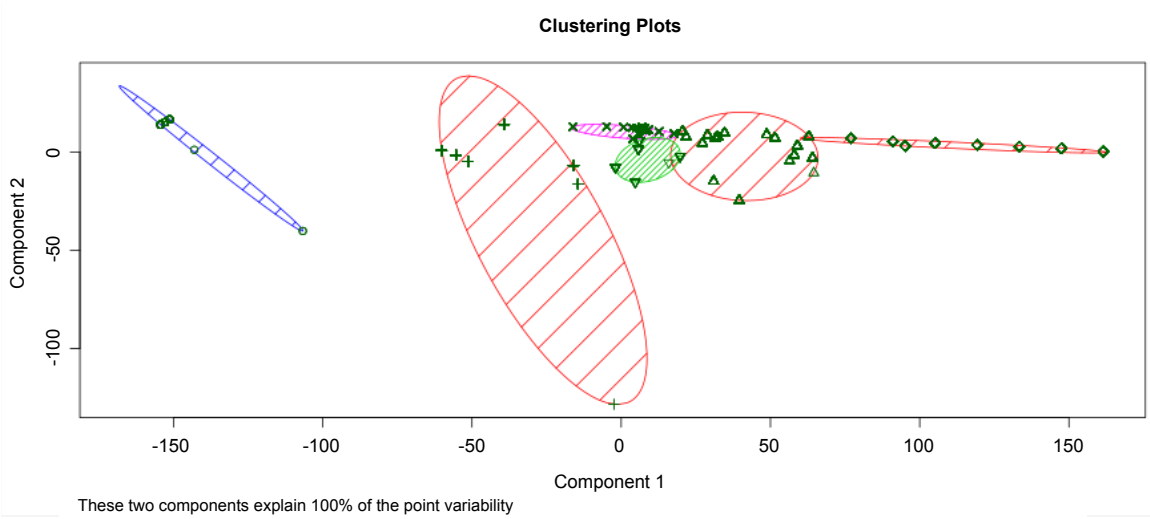

Figure 5: Discriminant plot.

Show 10 entries Search:

LSL	USL	Cluster	Frequency
-0.9	0.9	1	555
-0.2	0.2	1	6720
-0.2	1.5	1	9084
-0.14	0	1	239
-0.12	0	1	478
-0.1187	0	1	239
-0.115	0	1	239
-0.113	0	1	239
-0.1115	0	1	239
-0.108	0	1	239
LSL	USL	Cluster	Frequency

Showing 1 to 10 of 69 entries Previous 1 2 3 4 5 6 7 Next

Figure 6: Spec limits for the selected measurement unit.

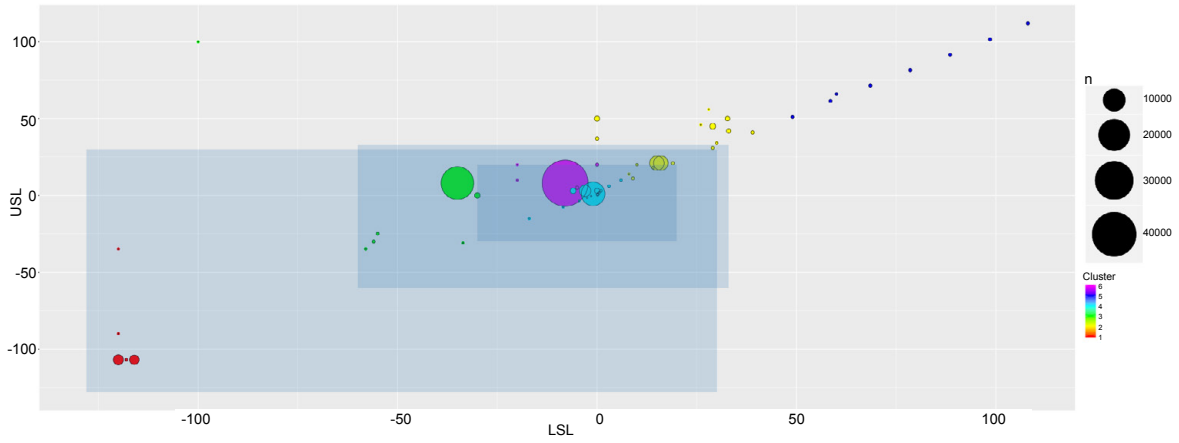

Figure 7: Test ranges, clusters, and capabilities.

database automated Gap Analysis is capable of creating an a priori list that identifies the test articles easiest to deploy onto a common test platform given the previous criteria.

Results

The method described in this paper saved a significant amount of engineering labor hours in the development of common test requirements. Test platform designers utilized the user interface to determine if the measurement was within range of the common test platform or if a gap modification was required. The results provided quantitative and visual support of test coverage, enabling common factory test platform designers to map products to test equipment across factories. This facilitated migration of existing factory configurations to common test platforms. We expect lower test costs from increased test platform utilization and a reduction in factory test platform heterogeneity over time.

The contribution of this paper is describing how an analysis can be conducted using a relatively small number of test parameters to provide critical information for design of common test equipment. In the current complex manufacturing environment, critical test platform design decisions are better informed, and made more quickly and confidently by quantified, data-driven support (see [16] for a description of the underlying team providing this support).

Conclusion

In conclusion, the construction of a database of historical testing data has become a foundational tool to develop a design for a common test platform. When starting any project to build common test positions, modular or composable testing systems for production, or conducting analysis for future development a similar system may prove to be an invaluable tool. This toolset has proven to save engineering labor, and early indications are this approach will reduce test cost by suggesting specific test platform commonality.

References

1. Nair C (2002) Modular test architectures for the aerospace industry. In AUTOTESTCON Proceedings-IEEE. Pp: 241-247.
2. Lohse JJ (2000) Common test architecture development for munitions level best platforms. In AUTOTESTCON Proceedings. Pp: 101-107.
3. Martel AW (1996) Test program structure and development on the Common Test Station. In AUTOTESTCON'96, Test Technology and Commercialization. Conference Record. IEEE pp: 86-94.
4. Breen MH (2000) Common test-platform family for module, board, and sub-assembly level test. In AUTOTESTCON Proceedings. IEEE pp: 79-84.
5. Schmitt W, Kelly M (1983) Army Digital Test Requirements Analytic Report. ManTech International Corp., Alexandria, VA.
6. Sonnenberg W, Sharpio A, Dewey M (2009) Consolidating test resources for avionics production test-requirements and applications. In AUTOTESTCON-IEEE. Pp: 389-392.
7. McLaughlin ME (2017) New product introduction at a technology company (MS Thesis, Massachusetts Institute of Technology).
8. Zhang Q, Karcher A (2012) System Reverse Engineering to Requirements and Tests. In Seventh International Conference on Systems, Saint Gilles, Reunion.
9. Kleinman J (1978) Status of technical manual specifications and standards. Technical Communication. 25: 24-26.
10. Neag IA, Seavey M (2007) Applications of IEEE P1671.1 ATML test description. In Autotestcon-IEEE. Pp: 197-204.
11. Haas R, Suzuki F (1987) Automated Test Requirement Document Generation. Rockwell International. Anaheim CA.
12. Mimbs SM (2007) Measurement Decision Risk - The Importance of Definitions. NASA.
13. Headrick WJ, Wong D (2015) ATE instrument function tracking. In IEEE AUTOTESTCON. Pp: 350-353.
14. Jain AK (2010) Data clustering: 50 years beyond K-means. Pattern Recognition Letters 31: 651-666.
15. Berkhin P (2006) A survey of clustering data mining techniques. In Grouping multidimensional data. Springer, Berlin, Heidelberg. Pp: 25-71.
16. Berryman S, Brio A, Burkhardt A, Ferkau, S, Gharbiah H, et al. (2017) Concept of operations for test cost analytics in complex manufacturing environments. In AUTOTESTCON, 2017 IEEE (pp. 1-8). IEEE.