

# Data-Driven Test Systems

In a data-driven test system, all product-specific information is stored in files. Within a product classification, the test software contains no product-specific information and does not have to be changed to test a new product. This concept lowers new product introduction costs.

by Adele S. Landis

In today's competitive marketplace, new products must be designed, manufactured, and delivered to customers in an effective and timely manner. This is the only way to maintain or create a desirable market share. At the HP Santa Rosa Systems Division (SRSD), the introduction of a new product from the research and development laboratory into a production environment is called the new product introduction cycle. This introduction cycle can be lengthy and expensive. One of the most critical and time-consuming phases in the new product introduction cycle is the development of the electrical test process.

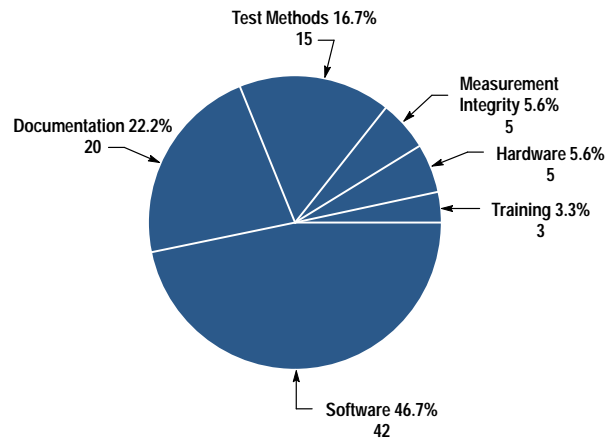
The electrical test process is that part of the manufacturing cycle that verifies that the product meets published customer specifications. The electrical test process is a combination of manufacturing resources, including:

- Test methods for doing each test that verifies a product's specifications
- Software that properly implements the tests
- The equipment necessary to perform the test methods
- Measurement integrity, which includes traceability, error analysis, and calibrations
- Documentation, including test procedures, test support documents, and the like
- Training of manufacturing personnel.

Fig. 1 shows an average distribution of the time spent on these resources for a group of electrical test processes recently developed at SRSD. The software portion of the process tends to be high because the division is moving towards completely automated testing.

Analysis has repeatedly shown that designing the test software by product classification rather than by individual products can result in large cost and time savings during the development process. A product classification can be defined as a group of products that have similar block diagrams and require the same type of testing. Some of the different product classifications that are manufactured at SRSD are network analyzers, spectrum analyzers, scalar analyzers, oscillators, amplifiers, and mixers.

Historically, production test software was written for and dictated by specific products or models. When test software was required for a new product or follow-on model, the test software was implemented or leveraged in a product-specific manner as shown in Fig. 2. This approach required continual introduction and/or reworking of the test software. The greater the number and variety of new products, the more time and software were needed. Also, this approach required

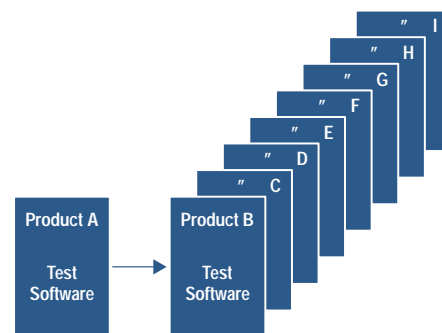


**Fig. 1.** Average proportions of time spent developing various manufacturing resources for a group of electrical test processes. Software dominates because the division is moving towards completely automated testing.

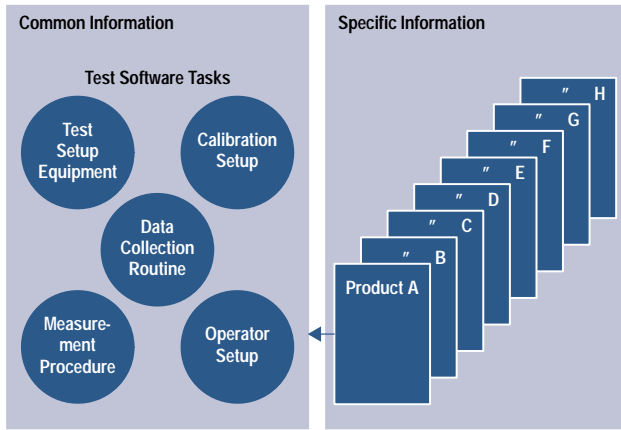
that an individual with software expertise be assigned for every cycle of new products that emerged from the lab.

## Data-Driven Software

Continued analysis has shown that by designing test software to handle large variations within a product classification, the overall development time of the electrical test process can be drastically reduced for all products within a classification. This greatly benefits both the manufacturer and the customer.



**Fig. 2.** Product-specific test software requires introduction or reworking of the software for each new product.



**Fig. 3.** In data-driven software, product-specific information is removed from the test software and stored in files. The software doesn't have to be changed to test a new product within the product classification.

A study of a large group of SRSD test programs across different product classifications was recently completed. The results indicate that all test software consists of certain types of tasks, including test equipment setup, calibration setup, operator setup, measurement procedures, and data collection.

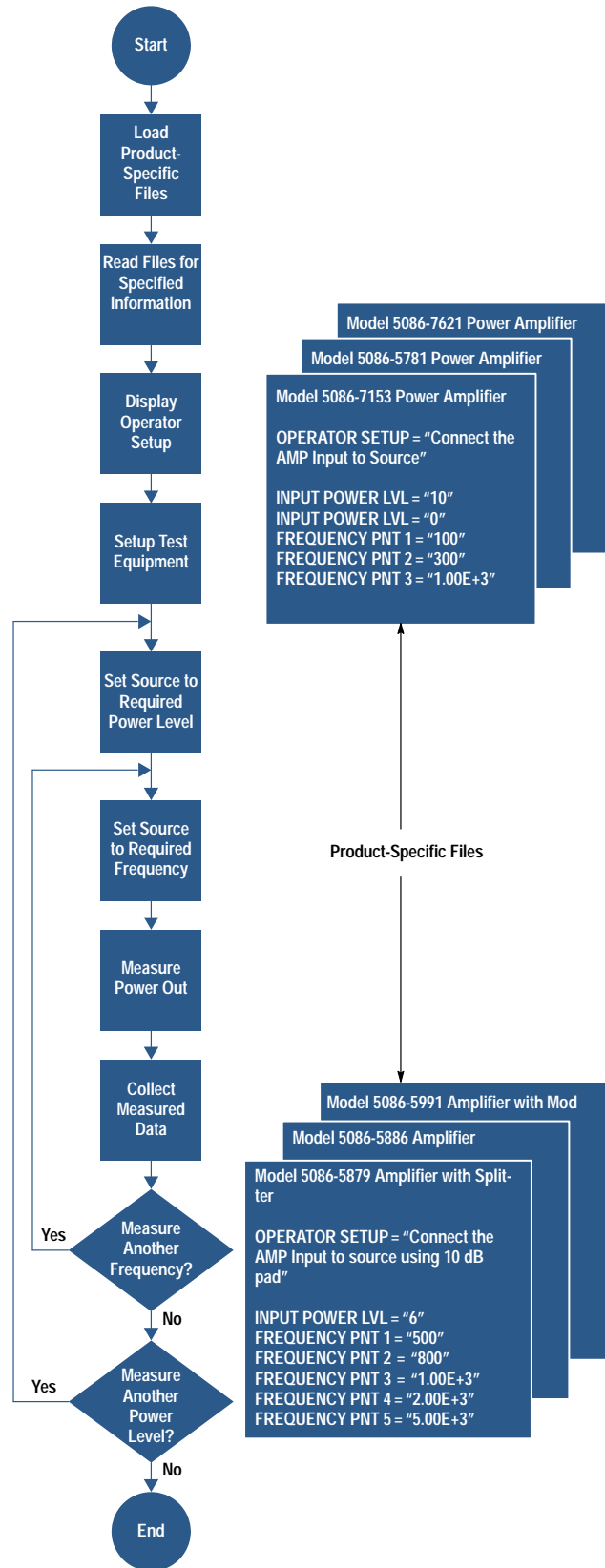
Regardless of which of these tasks are required for specification testing, all of these tasks can be broken down into two basic information types: common and specific. Common information is common to all products within the product classification. Specific information is specific to the individual products of that classification. Removing the specific information and placing it in external files leaves the common information to reside in the software (Fig. 3).

The separation of these information types results in static test software and dynamic product-specific files. The static test software contains the knowledge of the specific information required from the test's associated external file, the location of this file, how to load and read the file contents, proper measurement implementation, and the needed data collection routine. Since the test knows what information is required from its external specific file (but not necessarily the exact values), this file is considered defined.

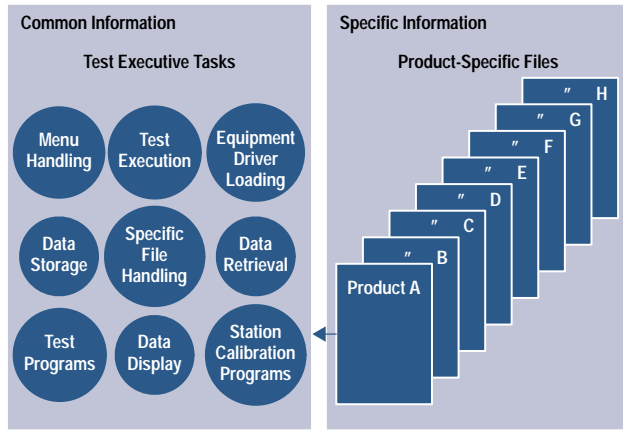
Test software that knows how to make the measurement, while the external files know where to make the measurement, can be called data-driven software.

Since data-driven test software is static, this method allows total test software reuse. Any product variation is handled in the external files. The building of the specific files can be automated easily since the type of information within these files has been previously defined by the test software. The specific file building is automated using a master form of the defined file and an editor. The editor only allows value entry. After the new values are entered into the master form, the product-specific file is stored. This approach, which does not require any software expertise, is extremely fast for new product setup.

A good example of data-driven software is an amplifier power test (see Fig. 4). The common information in this test, depicted by the software flow chart in Fig. 4, shows the tasks that are common to all amplifiers needing this type of test.



**Fig. 4.** Data-driven amplifier power test.



**Fig. 5.** Like data-driven test software, the data-driven test executive is made up of common and specific information.

The product-specific files contain the product measurement parameters and values required by the program to make the measurement. This same test software can be reused for any amplifier by building a product-specific file that contains the amplifier's measurement parameters.

### Test Executive

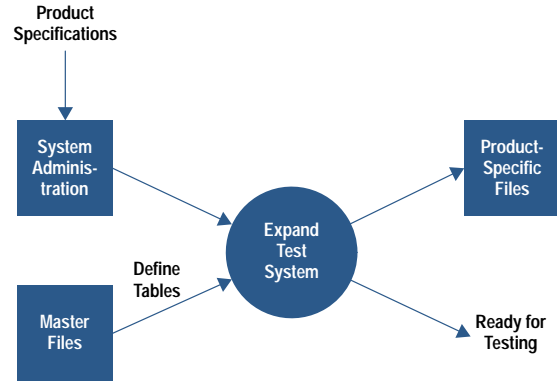
All manufacturers that are concerned with product test data need some form of a test executive. A test executive is a software package that handles the administrative and management responsibilities associated with test data. Even though the duties of the many test executives available vary depending upon user needs, all test executives have the same basic tasks as shown in Fig. 5.

Like the test software, test executive software is made up of common and specific information. By applying the data-driven theory to the test executive software, a data-driven test executive results. The test executive software then becomes static and any varying product information required by the test executive is available from external product-specific files. The building of these files can also be automated using master files and an editor.

The combination of data-driven test software and a data-driven test executive is a data-driven test system. The major benefit of a data-driven test system is that all the software within the test system is static. This benefit reduces the expensive and time-consuming cycle of product-specific software regeneration across a product classification. For test methods required by a new product that are not available in an existing data-driven system, the test method and associated test software must be developed. If the new tests are developed using the data-driven theory, then the tests will become part of the system's reusable tests and will be available for the next new product. Other key benefits are test system support and the automation of test system expansion.

### Expansion

A data-driven test system can be supported by a system administrator who resides on the production line where the test system is used. The system administrator can be a high-level technician who has some measurement and test equipment knowledge. Because the test system software is static, this



**Fig. 6.** Data-driven test system expansion.

individual does not need to have any software expertise. The system administrator's main duty is test system expansion.

Test system expansion is the process of adding a new product to a test system that already exists in the manufacturing environment. This process, whether for the traditional product-specific test system or a data-driven test system, can be broken down into two steps: adding the product into an existing test system and the initial product testing.

In the traditional product-specific test system, product addition takes approximately one day to eight weeks. This time frame is directly proportional to the amount of product-specific values that are imbedded in the test system software. Product addition in this type of test system must be performed by individuals with test software and test executive expertise. The time for product addition in a data-driven test system is approximately one hour and can be accomplished by individuals who have no test software or test executive knowledge.

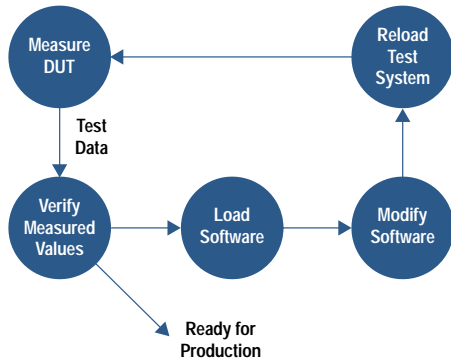
Expansion of a data-driven test system can be accomplished easily and rapidly with a programmatic procedure. The system expansion process is shown in Fig. 6.

The system administrator receives a new product and its specifications. Using a system expansion program, a set of master specific files, and an editor, the system administrator builds all the necessary external product-specific files required by the test executive and test programs.

### Initial Testing

After a product is added into a test system, the initial product testing must be done. This first testing is where the software used to measure the product specifications is verified to be working properly. In the product-specific test systems, this initial testing can be a very tedious process as shown in Fig. 7.

When the new product is measured for the first time and the measured values are not satisfactory or the test system stops because of a programmatic error, the software must be loaded into the computer, modified (debugged) and restored, and the test system reloaded. Then the product is measured again. Even though this process can be made faster using two computers, one for performing the actual measurement and the other for software debugging, initial product testing is



**Fig. 7.** Product-specific initial product testing.

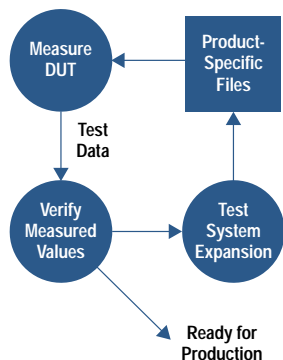
time-consuming and has to be performed by individuals with software expertise.

The initial product testing on a data-driven system is much easier, as shown in Fig. 8. After the product is added into a data-driven test system, the system administrator performs the initial product testing. If the measured values are not correct, any changes required are made to the external (product-specific) files through an editor immediately accessible from the keyboard. Once the system administrator is satisfied with the measurement, the product is measured a final time and the data is presented to the appropriate engineer for review.

This feature of a data-driven test system, which shifts the laborious work of measurement debugging from the test software to the product-specific files, is extremely beneficial. It not only allows nonengineers to perform the initial testing, but also, since proven test methods are being used, the overall initial testing process is very short.

### Adding Test Stations

All test systems whether old or new must be able to handle a variety of test stations. In the traditional product-specific test system, the system contains test-station-specific values imbedded in the software. With these station-specific values in the software, adding stations with extended capabilities or station duplication can be very time-consuming and a software expert is needed. This problem doesn't exist in a data-driven test system since the data-driven test system is station



**Fig. 8.** Data-driven initial product testing.

independent. These systems not only have the ability to handle numerous test stations with different measurement capabilities, but they also allow for test station duplication whenever production capacity is exceeded.

For extending a data-driven test system test station capability, the system administrator builds the new station hardware, assigns the station a new station number and/or name, and enters this information into the appropriate test system's station table.

Using this data-driven approach, SRSD has designed two complete test software systems as a pilot program. These data-driven test systems focus on measurement types within a product classification. These systems contain a set of well-designed tests that perform typical measurements required by a product classification.

### Mixed-Model Amplifier Test System

The mixed-model amplifier test system was the first system to validate the data-driven software theory. The system performs a set of basic tests that are required for complete amplifier characterization. The system has five different test stations: two network analyzer stations (40-GHz and 50-GHz frequency ranges) for complete s-parameter measurements and three scalar network analyzer stations (26.5-GHz, 50-GHz, and high-power 26.5-GHz frequency ranges).

This system began with two test stations with specific measurement capabilities. With the addition of more products, the need for stations with extended measurement capabilities arose. Since the data-driven system software is independent of test station equipment, we were able to add three additional stations with extended measurement capabilities without modifying any of the test system software.

This test system now provides complete characterization for over fifty separate products within the amplifier classification. Whenever necessary, the test system is expanded to incorporate new products by the system administrator. The total average product addition time is approximately three hours per product. The test process development time and expenses for these products have been greatly reduced.

For comparison, a product-dedicated test system used in microelectronic manufacturing was expanded for a new follow-on product. This system, which used some data-driven test software, still required one week of engineering time for complete system expansion because of the product-specific information imbedded throughout the test system software. Expanding this system for another product required three weeks of engineering/software time since the original tests could not be reused. This type of costly regeneration cycle will repeat itself again and again unless a data-driven test system replaces the current system.

The mixed-model amplifier test system design and set of tests required 487 days of software development time. Even though the initial test system design was very expensive, large savings occur each time a product is added to the test system. Test system expansion for a data-driven test system costs approximately 50 to 100 times less than the same expansion of a product-specific system.

The break-even point on the mixed-model amplifier test system was found to be four products, while the number of

products tested by the system is more than 50. The return on investment of the data-driven test system shows that this type of test system is necessary to stay competitive in today's marketplace.

### Vector Network Analyzer Test System

More recently, a data-driven test system was designed and implemented for a family of network analyzers. The complexity of instrument testing dictated that the data-driven test system be limited to a product family instead of a product classification. The system is currently testing five products. The new product introduction engineer estimated software generation time saved was approximately 500 engineering hours. More time would have been saved except that additional tests were required. These new tests were developed and added to the system in the data-driven format and are now part of the set of tests to be used by future family products.

In summary, by designing test software to handle large variations within a product classification, the overall manufacturing resources required for the electrical test development process are reduced over an entire product classification. This enormous savings occurs because the continual introduction or reworking of new product test software is eliminated.

### Acknowledgments

I would like to thank Claude Ashen for his support and guidance on structured analysis and system design, Steve Waite for his futuristic vision and at times blind faith that the test system would be completed in time to ship his products, and Wes Ponick for his enormous help in putting this project into words.

---

---

## Authors

August 1994

---

---

### 6 Scientific Graphing Calculator

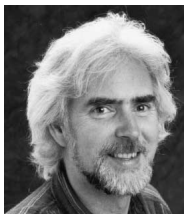
Diana K. Byrne



With HP since 1988, Diana Byrne is an R&D project manager for calculators at the Corvallis Division. She was born in Portland, Oregon and received a BS degree in mathematics from Portland State University in 1982, an MS degree in mathematics from the University of Oregon in 1987, and an MS degree in computer science from the same university in 1988. On her first HP project, she

developed software for plotting, graphics, and the EquationWriter for the HP 48SX calculator. She was R&D project manager for the HP 48G/GX calculator software and hardware. She is coauthor of a 1991 HP Journal article on HP 48SX interfaces and applications. Diana has two sons. Her favorite leisure activities are bicycling and reading.

Charles M. Patton



Software scientist Charlie Patton received a BA degree in mathematics and physics from Princeton University in 1972 and MA and PhD degrees in mathematics from the State University of New York in 1974 and 1977. He joined the HP Corvallis Division in 1982 as a software R&D engineer and has contributed to the development of the mathematics ROMs for the HP 75, HP 71B, HP 28C/S, and HP 48S/SX calculators. For the HP 48G/GX calculators, he worked on the RPL system and the 3D graphing routines, and was a general consultant and troubleshooter. He is currently working on several projects in the areas of software research, computer visualization, and symbolic and numeric techniques. He's also coauthoring a calculus textbook that incorporates technology as a teaching tool. He's named an inventor in three patents on operating systems and symbolic computation methods for handheld machines, and is the author or coauthor of technical papers on general relativity, representation theory, handheld computation, and calculus. He is a member of the American Mathematical Society, the American Association for the Advancement of Science, and the Internet Society. His outside interests include bird-watching, native plants, rafting and canoeing, fishing, gardening, Celtic harp, concertina, and calligraphy. A long-time HP telecommuter, he is also involved in remote sensing and internetworking.

David Arnett



David Arnett is a hardware design engineer at the Corvallis Division and has been with HP since 1991. He designed hardware and was the manufacturing liaison for the HP 48G/GX calculators. Currently, he designs hardware for the HP OmniBook product line. David was born in Cleveland, Ohio and attended Brigham Young University, from which he received a BSEE degree in 1989. He worked on avionics design at General Dynamics and on superconductivity research at Oregon State University before joining HP. He's a member of the IEEE. David is married, has two children, and enjoys music, both as a performer and as a conductor.

Ted W. Beers



Software R&D engineer Ted Beers came to HP's Corvallis Division in 1985 and has contributed to the development of the HP 48S and the HP 28C/S calculators. His work on the HP 48S includes the interactive stack, high-level display management, and user customization. He worked on user memory organization for the HP 28S and performed extensive testing on the HP 28C. For the HP 48G/GX calculators, he was responsible for the user interface elements. His work on a software technique for data and text entry has resulted in a patent, and he is the author or coauthor of three other technical articles, one written while he was in high school. Ted was born in West Lafayette, Indiana and received a BS degree in computer and electrical engineering from Purdue University in 1984. He's married and enjoys hiking with his wife and two beagles. He's also