THE 2ND GROUND-BASED SPACE FACILITIES SYMPOSIUM
17-19 October 2017, Toulouse, France

ENSURING DATA AVAILABILITY, RELIABILITY AND QUALITY ON COMPUTER SYSTEM FOR CONTROL OF LARGE "ARIANE" TEST BENCHES

Buth, Thorsten [(1)] and Stuchlik, Wolfgang [(2)]

[(1)] Werum Software & Systems AG, Wulf-Werum-Strasse 3, 21337 Lueneburg, Germany,
+49 4131 8307 142, Thorsten.Buth@werum.de
[(2)] German Aerospace Center (DLR), Institute of Space Propulsion, 74239 Hardthausen, Germany,
+49 6298 28 312, Wolfgang.Stuchlik@dlr.de

1 ABSTRACT

This article uses the example of computer systems for large-scale ARIANE test benches to show how data are reliably, safely and continuously acquired, stored and managed in order to have them ready for further evaluations that are reproducible and understandable at all times.

Computer systems for large ESA test benches have very distinctive characteristics. One of their essential features is, of course, that the individual sub-processes complete their tasks within very short time. But next to speed determinism is of equal significance. It means, for example, that redline conditions are monitored in exactly the given cycle and that command sequences are reproducibly executed at exactly the defined times in order to make the workflows at the test bench run the way the physical and chemical processes demand it. The actual capturing of data belongs under this kind of subtasks, and usually real-time operating systems are employed to accomplish them. Other jobs running in parallel, like data distribution, storage and visualization, are performed on customary computer systems as they are better suited for this kind of application.

Quality requirements put on such computer systems are really high as the test results of past decades must remain comparable. So, transparency and reproducibility are essential issues to be safeguarded for every test executed. The performance standards of the systems also show in the entire hardware and software components' availability throughout the year and in the guarantee that no test data can be manipulated or even lost.

It needs a special hardware and software design to afford such guarantees as all data stream paths, e.g. from the sensors or command/event interfaces all the way down to the archive file, need to be set up safely and probably even redundantly. This safety and, with it, the certainty that the entire digitalized data are fully available at all times is ensured by different internal layers of hardware and software functions. Some of those functions are based on standard technology and adapted for the specific purpose while others are specially tailored for a certain application scenario.

For DLR's test site in Lampoldshausen the company Werum Software & Systems in cooperation with partners has built several of such systems that fulfil the above mentioned requirements. The following article uses the solution for the P5 test bench as example to describe the system's intrinsic design principles. Apart from technical solutions, however, administrative aspects are of equal importance to ensure reliable operation over many years. For this reason, the second part of the article addresses such aspects and may also be regarded as "best practice" for successful projects.

2 ABOUT WERUM SOFTWARE & SYSTEMS

Werum Software & Systems AG is based in Lueneburg, South of Hamburg. With a workforce of over 120, it is one of the largest independent employers for IT professionals in Germany. For more than 45 years Werum has been implementing sophisticated software and systems for a worldwide base of customers, among them many renowned companies from the automotive and aerospace industry as well as scientific institutions and public authorities.

Werum's activities

Werum's activities focus on the support of customer-specific processes in the core areas of test data and information management, earth observation, eGovernment and enterprise information management. The software solutions are based on platforms specially developed for these areas.

Their aerospace-related activities started far back in 1993 with a measurement data management system for experiments of the STS-55 Space Shuttle mission, also known as D2 Spacelab mission. Also ESA's ROSETTA mission with the comet landing probe PHILAE was supported by Werum's software. Another focus is on systems for processing and handling satellite data, which are employed in many missions of European and international space agencies (e.g. ENVISAT, TerraSAR-X, Sentinel-1 & Sentinel-3 etc.).

Space engine test bench projects

This article, however, concentrates on systems for the execution of tests for space propulsion engines. In this area, too, Werum can look back on many years of experience in various projects. The first system developed for DLR was a corresponding system for their site in Trauen, Lower Saxony. Systems of markedly higher complexity followed in quick succession for many of DLR's research test beds in Lampoldshausen. The highlights of this cooperation so far are the Measurement, Control and Command systems (MCC) for the VULCAIN main engine of ARIANE 5 at the P5 test bench (since 2007) and a similar system for testing the upper stage of ARIANE 6 (since 2017) at the P5.2 test bench. The tasks and the typical setup of such a system are described in the next section.

3 BASIC MCC CONCEPT

As a general rule, MCC systems are employed not only for test execution with hundreds or even thousands of analogue and digital IO points, but also for the preparation and subsequent follow-up of tests. As a result, there is a set of most diverse tasks an MCC system has to fulfil:

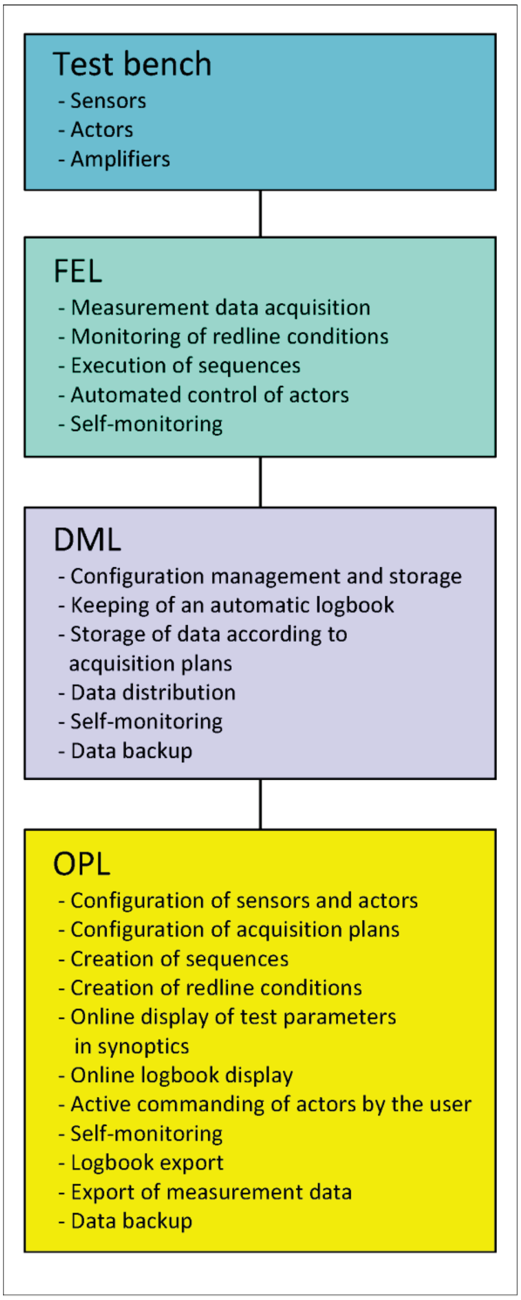
- Configuration and management of sensors (PA, PD, PR, V, TC, Pt, D)
- Configuration and management of actors
- Configuration and modification of hardware limits (cabling and amplifier types)
- Validation of data base entries
- Configuration and management of acquisition plans
- Creation of programs and sequences
- Creation of redline conditions
- Configuration management and storage
- Measurement data acquisition
- Creation of measurement cross sections
- Keeping of an automatic logbook
- Storage of data according to acquisition plans
- Data distribution (measurement data, logbook messages, status and commands)
- Monitoring of redline conditions
- Execution and suspending of programs and/or sequences on demand
- Automated control of actors
- Provision of a standardized time base for data storage, sequence execution and redline monitoring
- Online display of test parameters (e.g. measurement data or valve positions) in synoptics
- Online logbook display, logbook freeze function and string searching by SQL instructions
- Online trend display for preselected channels
- Active commanding of actors (e.g. valves) by the user
- Self-monitoring of all relevant MCC components
- Logbook export for defined time range
- Export of measurement data in different file formats for external analysis tools
- Data backup

These tasks are distributed across several computer systems that may vary considerably and are assigned to different levels:

- Frontend level (FEL)
- Data management level (DML)
- Operator level (OPL)

Distribution makes it possible to choose the computer system that is suited best for the task at hand. The tasks of measurement data acquisition, sequence execution, redline monitoring and actor control, for example, are executed on special real-time systems, so-called frontend systems, in order to ensure that the necessary cycle times are observed. As opposed to this, it's customary PC systems that can show their strengths when it comes to online displays at the operator terminals. For data storage and distribution, special server systems are used that are optimized for high data throughput rates and also run database systems, for example.

Some of the tasks mentioned can be assigned to individual computer systems while others call for an interoperation of all levels. Fig. 1 shows the levels the different tasks are primarily assigned to.


Test bench

- Sensors
- Actors
- Amplifiers

FEL

- Measurement data acquisition
- Monitoring of redline conditions
- Execution of sequences
- Automated control of actors
- Self-monitoring

DML

- Configuration management and storage
- Keeping of an automatic logbook
- Storage of data according to acquisition plans
- Data distribution
- Self-monitoring
- Data backup

OPL

- Configuration of sensors and actors
- Configuration of acquisition plans
- Creation of sequences
- Creation of redline conditions
- Online display of test parameters in synoptics
- Online logbook display
- Active commanding of actors by the user
- Self-monitoring
- Logbook export
- Export of measurement data
- Data backup

Fig. 1: Tasks assignment

This way of distributing tasks inevitably involves a modular overall system which ensures that the impact future changes and extensions have is limited to only few, directly affected components.

The distribution of tasks across several systems can also be utilized to increase fail-safety. Moreover, the presence of several operator terminals, for example, makes it possible for different users to work in parallel. Last but not least, the use of several computer systems allows for load balancing, thereby increasing the overall system's scalability.

Wherever possible and reasonable, standard hardware and software components are employed. As a result of this, costs are reduced, the systems are open to meet future needs and single-supplier dependency is minimized.

The components of the MCC are linked by powerful networks. Fig. 2 shows the basic setup of an MCC including the major data flows between the levels.

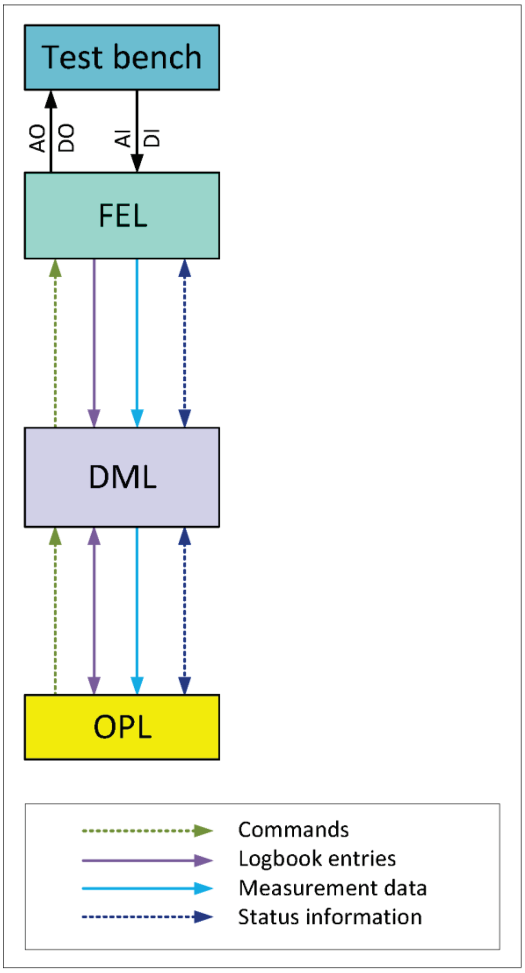


Fig. 2: Basic MCC setup

4 ERROR HANDLING

Important factors for the reliability and availability of a MCC certainly are robustness and a high quality of the hardware and software components.

Self-monitoring

Equally essential, however, is the system's self-monitoring functionality that ensures that error states are detected unfailingly. It's the only way to have automated error handling processes started or to have the user informed about the presence of an error so he or she can take a decision and respond adequately.

After all, a time-consuming and costly test run should under no circumstances be started with a deficient system that entails the risk of having to terminate a test or of jeopardizing the safety of humans, the test bed or the unit under test.

In case of an error

If an error occurs in a test run, the desired automatic error handling procedure usually is the automated execution of the stop sequence that puts the test bed and the unit under test into safe condition. The MCC might still be able to do it on its own – provided its sequence generator still works properly. As it's not possible to guarantee that for any conceivable error scenario, MCC systems are frequently extended by adding a so-called Emergency Stop Systems (ESS) which presents the last fall-back level to accomplish a safe shutdown.

An ESS is triggered by the absence of a heartbeat signal which the MCC sends to the ESS while in normal operation. When designing an ESS, it's advisable to pursue a hardware and software concept that is at variance with that of the MCC to prevent that the same error can occur in both systems.

5 REDUNDANCY

The availability of an MCC can be increased by a redundant component structure. In an ideal setup, the backup system seamlessly takes over in case of an error, thereby enabling successful completion of an already started test. The definitions to be made first in laying out the system, are the desired degree of redundancy and the specific components to be redundant. In case of the MCCs for P5 and P5.2, it was the data management level that was implemented redundantly. On operator level, redundancy is constituted by the large number of equivalent operator terminals so that any failure of a terminal can easily be compensated for. Fig. 3 illustrates the redundancy concept of the P5 test bed with its so-called redundant DML halves.

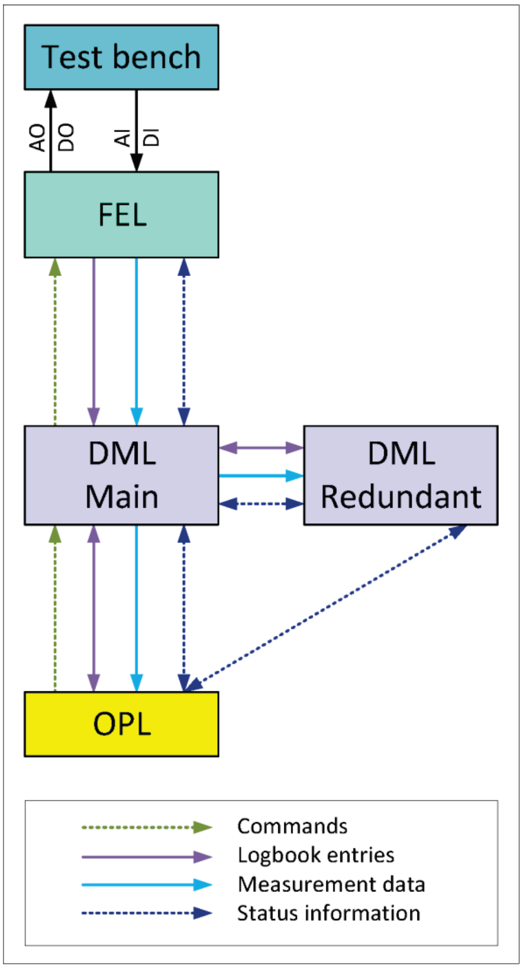


Fig. 3: MCC with DML redundancy in nominal operation

When designing an MCC, redundancy should be considered from the very beginning as it does not suffice to just let MCC components run in parallel. Not only must the redundant MCC components monitor themselves, they also have to exchange status information and take the correct actions in case of an error.

If subcomponents within one DML half fail, identifying the correct actions is rather easy. So, when the database system on the main side fails, for example, operation altogether switches over to the backup system. If it is the database system on the redundant side that fails, however, the backup system will be marked as being unavailable in order to prevent any switchover to the unhealthy side in the sequel.

Communication problems on status or data connections between the two halves pose a major challenge in identifying adequate response as they may in principle have different causes:

1. Problems on the main side - (e.g. damaged network card)
2. Problems on the main side - (e.g. complete failure of the side)
3. Problems in the wiring - (e.g. damaged cable)
4. Problems on the redundant side - (e.g. damaged network card)
5. Problems on the redundant side - (e.g. complete failure of the side)

In case of no. 2, a switchover to the backup system would definitely be reasonable, while in case no. 5 trying to switch over to the unhealthy backup system would certainly be the wrong decision. In the cases no. 1, 3 and 4 the problem is confined to the communication between the two halves and either half on its own is functional. The difficulty is that it sometimes is impossible for a communication partner to identify the reason for the communication problem without doubt or within the necessary period of time. And, what is more, the error has an effect on both sides and both sides consequently must respond in some way. Therefore, it is essential to develop a suitable logic that ensures that the responses are compatible so as to maintain proper working order for at least one half and to have no incorrect conclusions drawn

The solution for the P5 MCC is the following: only the half active at a time maintains a connection to the FEL. The data received through this connection are mirrored from the DML Main (DMLM) to the DML Redundant (DMLR). The switchover to the backup system in case of an error is always triggered by the backup system. The main system can only solicit a switchover from the backup system. This is the reason why the two DML halves continuously exchange status signals via a dedicated network connection which is independent in terms of hardware.

Should the main side detect an error in the communication with the DMLR (e.g. the non-appearance of a status signal), the DMLM will assume the DMLR to be unhealthy and abstain from requesting a switchover to the DMLR from that point. The problematic connection will consequently not be used any longer, no matter which of the above events occurred. As far as the DMLM considers itself to be healthy, it will continue performing its tasks as before.

If the DMLR detects a connection problem and regards itself as healthy, the switchover will be carried out. Here, too, the problematic connection stops being used. Instead, the DMLR sets up its own connection to the FEL, thereby cutting the connections between FEL and DMLM. This disconnection also has the effect that the DMLM – despite the missing status connection – is indirectly informed of the DMLR's taking over and changes to the "unhealthy" status. The situation after a switchover is depicted in Fig. 4.

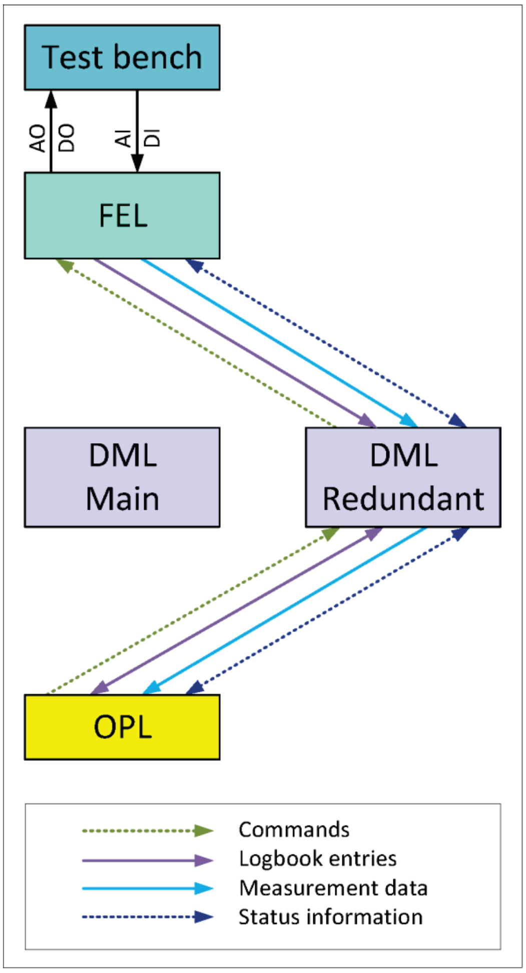


Fig. 4: MCC with DML redundancy after switchover

Another important aspect in designing a redundant system is command control. Commands can be sent by the user at an operator terminal. The DML receives the commands and passes them on to the FEL. Should it happen that a switchover takes place at the precise moment the command is sent, the command must be certain to reach its destination, but it must reach it only once. To visualize possible consequences if it didn't just imagine what would happen if a valve was erroneously toggled twice or a sequence was started twice.

6 DATA ORGANISATION AND BACKUP CONCEPTS

The demand for reliable operation over years with reproducible and always comprehensible results puts special requirements on the handling and the organization of data. The term "data" in this context is a rather general term and it's worth considering the different categories of data.

First and foremost, it's the integrity of software components that is important for reliable operation. Among the components relevant to ensure it are the operating system installations on the various computer systems, the specific MCC application software and the associated configuration files. Within the scope of maintenance activities, backups of those components are created periodically in the form of images. If it comes to the worst, they can be used to restore a system in a minimum of time.

A suitable configuration management suits to fulfil the demand for reproducibility. Special databases or commercial version control systems are employed for this purpose. Next to sensor, amplifier and actor parameters they also accommodate items like sequences, redlines and information on test runs. This approach makes sure that a test can be repeated at some date in the future with the same configuration. The configuration management is organized in a three-level tree structure that is divided into campaigns with configuration revisions arranged below them and the test runs below the configuration revisions.

One of the tasks the configuration management is co-responsible for is traceability, which ensures that even after years, sensor parameters or sequences, for example, can still be viewed and allocated to a specific test run.

The measurement data of the MCC are stored as raw data in archive files together with sensor and amplifier parameters. The set of data forms an inseparable unit that admits of independent evaluation. As the data stored are raw data subsequent re-processing is possible, e.g. if a sensor calibration turns out to be invalid.

The logbook is implemented by a special database. Consequently, database tools can be used to browse it.

The data storage is located on the DML servers. They are equipped with commercial RAID systems which automatically duplicate the data and mitigate the impact of a possible hard disk crash. This ensures safety of the data as well as smooth continuation of operations.

The P5 MCC setup includes redundant DML servers which store the data in parallel. This way, the data are duplicated twice in normal operation.

Automatic tools are available to perform the periodic backup of user data (configuration data, database contents including logbook and measurement data files). They can transfer the data to a commercial NAS system located in a different fire compartment.

Over the years, terabytes of data accumulate in operating the MCC – especially due to high frequency acquisition. In order to release the MCC resources for new test runs, campaigns can be selected to transfer the associated measurement data and logbook entries to an external system, such as an NAS system. In the event that those data are needed in the MCC again at some other time, they can easily be re-imported using the available MCC tools.

7 LONG-TERM CONSIDERATIONS

The article at hand so far attended to the technical aspects creating the basis for a stable system of high availability and reliability. When it comes to operating a complex hardware and software system for many years, though, there also are organizational aspects that need to be considered. In particular, it's more than helpful if the MCC manufacturer has been established in the market for many years and there is a fair chance that they can support the product's lifecycle for quite some time.

In the following, we go into the details about aspects and measures that have contributed significantly to the P5 MCC project's success of many years.

Spare parts

MCC systems in general consist of multiple computer systems. With the P5 and P5.2 MCCs they amount to a total of about 30. This number clearly indicates that single failures are to be expected sooner or later when running a system for many years. Therefore, it's advisable to lay in an ample stock of spare parts so that the components concerned can be replaced or repaired at any time. The endeavour is to use high-quality standard components for the computer systems. But for the FEL real-time systems in particular, very specially geared hardware components are employed which may have a lead time of many weeks; a circumstance that illustrates how important it is to have spare parts in stock.

Another reason for holding stocks of spare parts is potential obsolescence (see below) which may mean that certain spare parts are no longer available on the market at all. Often enough there are alternative products to use, but they may entail a modification of configurations, for example, or the installation of different drivers. With spare parts of the same type in stock this kind of trouble can be avoided as the components can be replaced simply by plug & play.

A further advantage available spare parts offer is that components can be exchanged on a trial basis, which is a suitable approach to narrow down error causes when there is an unclear error situation.

Maintenance

Similar to modern cars, advanced computer systems require only little maintenance, but are not free of it. Regular maintenance ensures that reliable operation of a system with the accustomed qualities is possible for many years. For maintenance, the individual components are inspected thoroughly under a series of different aspects:

- Visual inspection of the components
- Inspection of the operating system's log files
- Inspection of the application software's log files
- Abnormal development of noise with fans or hard disks which could point towards imminent failure
- Checking for and removal of dust, also inside the computer housing
- Checking of hard disk capacities
- Checking of IO board calibrations
- Checking of spare part stock

The results of the maintenance operations are summarized in a report that also makes recommendations for further actions.

Obsolescence

Especially in the world of computers technology is moving fast, which holds true for hardware, software and (logical) interfaces between components. It frequently happens, for example, that computers and screens are no longer available in exactly the same design after just one year, but are replaced by new types. This does not pose any problems as long as the successor model is fully compatible.

New computer models, however, often support only current operating systems because the required drivers of new hardware components, for instance, are no longer developed to work with older operating systems. If a computer needs to be purchased because the old device is defective or a system extension is desired, the fact that the application software is unqualified for the new operating system may pose a serious problem.

The example suits to emphasize how important obsolescence management is if an MCC system is supposed to run reliably for years or even decades. The first step in establishing obsolescence management is to observe the market in order to spot possible cases of obsolescence in the first place. Based on the findings, suitable actions can be defined and implemented in the next two steps. Some possible measures are:

- Replenish spare parts stocks as long as the parts are still available
- Integrate successor models on a trial basis to be prepared for emergencies
- Exchange all components concerned, which frequently is attended with a gain in performance in case of computers
- Design a new setup for the subsystem concerned that does not need the part affected by obsolescence

The first three measures have already been implemented successfully in the P5 MCC project, depending on the specific situation. Owing to the design, it has not yet become necessary to create a completely new concept for any of the subsystems.

Obsolescence management is a continuous process. Necessary actions should not be deferred too long as, otherwise, there will be a serious backlog in technology and investments. So, regular replacement of computer systems is advisable not only to guard against the risk of failures due to advancing age, but also to prevent operating systems from lagging generations behind and to profit from technological progress.

Continuous improvements

As a general rule, the demands made on a test bench vary over the years. Especially with test benches for research and development purposes it often happens that requirements on channel numbers are added or devices with new interfaces that should be connected. Here, it definitely pays off if the MCC, too, has a modular and scalable design. In the past 10 years, the P5 MCC underwent several extensions and was brought up to date so that it can now be used for the VULCAIN 2.1 engine of the new ARIANE 6.

But it's not only the major extensions that are vital in ensuring that the test bench and the MCC meet varying requirements at all times. Improvement management, too, is a continuous process that is of equal relevance and requires regular communication between customer and manufacturer. A suitable way of handling it is to hold periodic meetings at which both parties can swap ideas and views on latest developments, requirements, experiences or technical opportunities. The information gathered can then serve as basis to deduce suitable planning steps.

As a matter of fact, though, improvements are only one side of the coin. There is a high likelihood that errors occasionally occur with systems as complex as the MCC. They may be errors of the actual system, errors caused by connected external systems or errors caused by operating mistakes or a lack of understanding. What all these error categories have in common is that immediate and direct contact to the manufacturer is indispensable in order to start error analysis and repair right away. The best way of tackling it is to conclude a maintenance contract which makes it possible to set to work without delay and without the need for either side to clear administrative or other hurdles first.

Training

Intensive training usually is integral part of the overall project for a complex computer system. Even with supposedly simple applications like text-processing systems only a fraction of the dearly paid functions is used - depending on the user's level of knowledge. Complex MCC systems are designed to perform expensive test runs in a manner that the safety of humans and machines is protected. Therefore, efficient mastering of the systems is essential for reliable operation and expenditure for a well-trained team should not be regarded as a cost factor but as an increase in the investment's value.

For various reasons, though, users should be trained not only during the commissioning phase but also afterwards whenever the need arises. The number of persons operating large test benches over time is large and staff turnovers are commonplace. So, new team members should be imparted a similar level of knowledge as experienced team members have it.

For experienced staff, too, training courses may prove beneficial as they can brush up or deepen their knowledge. Many questions or the desire to go more into details don't manifest until first working experiences have been gathered with a new system and then it's a good opportunity to have an expert available to discuss the issues with.

8 SUMMARY

This article explains the basic design principles of the MCC that has been operated successfully at the P5 test bench at the DLR test area in Lampoldshausen for many years. The second part goes into the details about some organizational aspects which, strictly speaking, apply to any hardware and software system of high complexity. But especially when reliable operation of the overall system must be guaranteed over a number of years, these factors are crucial next to a solid technical solution.

9 ACRONYMS

AI	Analogue Input Channel – view from the MCC
AO	Analogue Output Channel – view from the MCC
D	D sensor type - strain gauges, supplied by voltage
DI	Digital Input Channel [e.g. event from the test bench] – view from the MCC
DLR	German Aerospace Center
DML	Data Management Level
DO	Digital Output Channel [e.g. command or trigger pulse] – view from the MCC
ESA	European Space Agency
ESS	Emergency Stop Systems
FEL	Frontend Level
IO	Input Output Interface points
IT	Information Technology
MCC	System for Measurement, Control and Command
OPL	Operator Level
PA	Pressure Absolute
PC	Personal Computer
PD	Pressure Differential
PR	Pressure Rapid
Pt	Platinum element supplied by constant current
RAID	Redundant Array of Independent Disks
TC	Thermocouple
V	Vibration sensor