Nationaal Lucht- en Ruimtevaartlaboratorium
National Aerospace Laboratory NLR

NLR-TP-98135

Avionics development in a concurrent engineering environment, from virtual prototyping to testing

L. J. Aartman, L.C. Eveleens and S. Wellink

Nationaal Lucht- en Ruimtevaartlaboratorium
National Aerospace Laboratory NLR

NLR-TP-98135

Avionics development in a concurrent engineering environment, from virtual prototyping to testing

L. J. Aartman, L.C. Eveleens and S. Wellink

This report is based on a presentation held on the 5th European Concurrent Engineering Conference organised by the Society for Computer Simulation, Friedrich Alexander University, Erlangen-Nuremberg, Germany, April 26 - 29, 1998.

Division:	Electronics and Instrumentation
Issued:	July 1998
Classification of title:	unclassified

Summary

Avionics product development is a multi-disciplinary activity carried out by multiple design groups and often distributed over several companies. The avionics products are usually composed of multiple hardware and software elements. By improving system specification and requirements verification techniques, early shortcomings, - thus future problems - can be avoided. Early verification therefor is important as a general quality objective and early prototyping has become an important means to achieve this.

System modeling techniques are being applied to create virtual prototypes for verification by simulation in early stages of the development. More and more an ‘enter-once-use-many' strategy is actively pursued. By maintaining a coherent product interface specification throughout the various steps in the development of (sub)system components a high quality level can be achieved. At the National Aerospace Laboratory NLR an avionics prototyping environment ARTIS (Avionics Research Test and Integration System) has been developed to support both virtual, hybrid and physical prototyping. ARTIS fulfills the exchange of data between the various up-to-date components of the system under development and allows for a simulation of the most recent design. Examples of the use of this approach in the development of various avionics products (like cockpit panels and airborne data management equipment) are presented. The present verification environment is based on PC-platforms and Windows software for testing, and uses the I-Logix's StateMate tool for modeling.

Contents

7 Figures

(18 pages in total)

1 Introduction

‘Avionics' is a word derived from the combination of aviation and electronics and is used to mean any system in an aircraft which is dependent on electronics for its operation, although the system may contain electro-mechanical elements (Collinson, 1996).
Avionics product developments are multi-disciplinary activities carried out by multiple design groups and often distributed over several companies. Avionics products are usually composed of several hardware and software components and differ in many ways from ground based equipment carrying out similar functions, in particular when it concerns reliability, safety and integrity. Today’s projects are characterized by increasing complexity (both in terms of the product and in terms of the development process), while development time shortens and budgets shrink. High quality must be maintained throughout the complete product development cycle. Validation of product requirements is performed during the qualification process, which involves extensive analysis and (environmental) testing. It is almost impossible to perform avionics product developments without applying concurrent engineering design principles.

Product effectiveness (the capability of a product to meet customer objectives) is among others a function of availability, reliability and capability. In terms of product life cycle costs, 40 to 60% of the total costs may be committed as a result of decisions and actions taken early during development, where only 3 to 5% of the total development and production cost are expended. (Pecht 1995).

Being aware of this, an elaborated performance requirements specification and verification has to be performed during the proposal phase. This is the time to review and negotiate contractual requirements; not only performance requirements (operational, physical or functional characteristics) need to be analysed, but also requirements on reliability, testability, maintainability and safety have to be assessed. The link between reliability and testability has great impact on the design and development cost of avionics, as well as on the product cost. A significant effort is required to implement Built-In Testability (BIT) components and to demonstrate or analyse product performance in the fail state of the avionics product. By improving system specification and requirements verification techniques, early shortcomings leading to future problems can be reduced or avoided. Early verification therefor is important as a general quality objective and early prototyping has become an important means to achieve this. Today more tools become available to support early prototyping. At the National Aerospace Laboratory NLR an avionics prototyping environment is under development that supports a coherent virtual, hybrid and physical prototyping.

1.1 Virtual prototyping of avionics

Virtual prototyping is based on computer simulation of the product model. For avionics this simulation can be in terms of both the functional and operational performance before a final selection of implementation technology is done. However, often a preliminary allocation of hardware and software components is performed already during early system design stages or is even prescribed by the customer. To avoid that hardware and software requirements disjoin, hardware and software co-design is an important approach during avionics development.

High level modeling of avionics at NLR is done with the StateMate tool of i-Logix (i-Logix 1996). In the StateMate approach, the descriptions used to capture the system specification are organized into a functional (what?), behavioral (when?) and structural (how?) views. The functional and the behavioral views describe the conceptual model (close to the problem domain) where the structural view describes the physical model (close to the solution domain). Each view has its own graphical modeling language:

- Activity-charts for the functional view
- Statecharts for the behavioral view
- Module-charts for the structural view.

Block diagrams can represent both function and structure. All these high-level concepts apply equally to hardware and software development processes. (Harel et al. 1990).
Additional non-graphical information related to views themselves and their interconnections is provided in the Data Dictionary (Harel and Politi 1996).

Steps to be taken during this system design process are (Sahroui et al.1996).:

- multi-formalism specification;
- iterative partitioning and performance evaluation;
- architecture prototyping.

As soon as a functional and structural baseline is established also the non-success (fail) states can be analysed and derived requirements for adequate measures in the field of reliability and testability can be defined.
This modeling technique is used to support the following activities:

- early requirements verification. Understanding the system design, avoiding costly errors later in the design process;
- requirements traceability;
- automatic code generation, to assist in the implementation of software and hardware (e.g. by means of synthesis into the hardware design language VHDL);
- define test system boundaries and test specifications.

At NLR this latter approach is supported with a tool called ARTIS (Avionics Research Test and Integration System ARTIS). Together with a coherent product interface specification ARTIS can provide the development team with a mix of virtual and physical prototypes of system (components) under development throughout all phases of the development (from requirements analysis to test and integration).

1.2 Avionics Research Test and Integration System ARTIS

ARTIS was designed to support the test and integration phase of an avionics (sub)system, and behaves like an operating system, which is optimized for test applications. A (sub)system Unit Under Test (UUT) is connected to a target environment which consists of:

- a workstation (or PC) running the ARTIS-kernel,
- hardware interfaces which physically connect the UUT with the workstation and
- a number of software modules which describe the target environment of the UUT.

Not only hardware components, but also software modules can be included in the test configuration. This is done for two reasons: either the physical equipment is not yet available, but a simulation model (virtual prototype) of a system component is, or a complex target environment is simulated for ease of use. ARTIS fulfills the exchange of data between the various components of the system under development and allows for verification by a combination of simulation and test of a specific design configuration. See figure 1 for the ARTIS architecture.

In order to maintain integrity throughout the design cycles of the components of the system, a strong interface has to be maintained throughout all phases of development. Interface signals are defined and captured at the context level of the product. An Interface Control Document or an Interface Requirements Specification is used to capture the signals and form the basis for the interface tables to be used by the simulated test bus. Stub-components describe the operator interfaces of the target system, and are used to inject test stimuli in the integrated system and to monitor responses. This is done either manually, via a user interface panel that is associated to a stub-component, or automatically, by executing a test-sequence. When real test-equipment is introduced in an ARTIS environment, equipment stub drivers replace the stub components.

ARTIS has been used successfully in the establishment of a complex test environment for an Airborne Data Link Processor and for an avionics radio frequency indicator unit.
At NLR tools are being developed to automatically export models developed in StateMate to ARTIS software components. A conversion program extracts external signals from the model and creates a control file for incorporation into the verification system.

Besides a seamless integration of all components (through Dynamic Link Libraries) ARTIS provides for the following features:

- monitoring and control of test-bus
- test operator user interface
- replay function of recorded data
- automated test script execution
- report generation.

2 Examples

During the last two years the individual steps have been introduced in the framework of several projects, leading to a better understanding of the desired coherent design environment. Examples of the use of combining virtual prototyping and testing in the development of various products (like cockpit panels and airborne data management equipment) are presented.

2.1 Virtual Prototyping of a fuel system

Based on a preliminary requirements specification, a system design for a new helicopter Fuel System was modeled using StateMate. The GUI was created with LabWindows/CVI for both visualizing the frontpanel and to visualize the simulated behavior of externally connected equipment like sensors and valves. Since ARTIS was not available at that time, exchange of signals between the different elements was done with a simple interface program (see figure 2). All components were combined into one executable program that could be handed over to the customer for early verification of the virtual prototype (see figure 3).
This prototype demonstrator is among others used to check the understanding of functional requirements in a virtual reality environment.

2.2 Fast-prototyping of an avionics indicator unit

One of the projects where modeling and testing was applied, was the development of a frequency indicator unit for a helicopter. Within this development project three relevant stages for prototype verification and testing were identified (see figure 4):

1. functional modeling and virtual prototyping in a simulated environment;
2. verification of physical (critical) components in conjunction with elements of the virtual prototype in simulated environment, a so-called hybrid situation;
3. Verification of the hardware within a test environment based on elements of the simulated environment.

During the requirements analysis of this LED-based indicator unit the preliminary functional design was modeled in StateMate for simulation as a first step. This indicator design existed of two major parts:

- external interface and controller;
- display interface and the displays.

The external interface consists of ARINC 429, lighting control signals and power lines. The LED-based dot-matrix displays (two lines of seven characters each) are controlled by a chipset, one chip per character. All these chips are connected in serial and controlled by data, clock and load lines. The controller is a single chip microcontroller running embedded software for data

management, display control and Built-In Test functions.
By developing this model, a number of omissions in the development specification were discovered during simulation and could be corrected in an early stage. The model also contained elements simulating other externally connected devices (simulated environment) and graphical user interface elements. To improve our understanding of a newly applied display controller chip, it was modeled in more detail.

Also in a very early stage of the development an ergonomic model, based on the real displays, had to be delivered to the customer for man-machine interface evaluation. It was decided to provide an empty unit, with only the front panel, containing the displays, controlled by a Personal Computer (PC) using the parallel printer port as the physical interface, delivering the above mentioned required signals. During this second step, the display part was separated from the model and a driver for the physical parallel printer port of the PC replaced the interface between the display interface and the displays. The graphical user interface elements were ported to the LabWindows environment, as a preparation for the final physical prototype test, and C-code was generated based on the StateMate model description. After compilation of the model and its simulated environment, it became available as an executable program that was able to control the physical display elements via the parallel printer port.

In the third step the complete physical prototype is controlled in a hardware environment, using the existing Graphical User Interface (GUI) elements available from the previous steps. Now the above-mentioned ARTIS kernel was introduced to connect several hardware interface device drivers together in the physical prototype test environment. In the Interface Control Document (ICD) each electrical signal or parameter in a data stream has an identified entry, describing its characteristics like source, destination, type, resolution etc. These ICD entries are translated into interface tables to be used by the ARTIS kernel to transfer the data as required and requested. During this third step the GUI is exactly the same as in the second. Within the same executable there is still an option to connect an individual display unit to the parallel printer port.

2.3 Airborne Data Link Processor

ARTIS has been successfully used to test the Airborne Data Link Processor (ADLP) that has been developed by NLR. The ADLP provides several communication protocols (i.e. broadcast and X25) via the Secondary Surveillance Mode S transponder of aircrafts.

The ARTIS test configuration simulated both the aircraft environment: X25 DTEs, the data broadcasted by several aircraft systems and Mode S Specific protocol-devices, as well as the ground-environment: the radar-interrogator, the Ground Data Link Processor (GDLP: the peer protocol-layer of the ADLP) and the ground X25 DTEs (see figure 5).

The models in the ADLP-environment were coded in C. The interface boards which provide the required avionic ARINC-interfaces and protocols (i.e. ARINC429, ARINC718) were mounted in a VMEbus based rack which was connected to the workstation running the ARTIS-kernel and the models in the ADLP target-environment (see figure 6).

The EUROCAE Minimum Operational Performance Specifications (MOPS) document for Mode S Aircraft Data Link Processors was semi-automatically translated into test-scripts for execution in the ARTIS-kernel. These scripts were used for automatic and formal testing of the ADLP and for the generation of the test-reports.
The present verification environment is based on PC-platforms running Windows95 in conjunction with the LabWindows/CVI software.

3 Future plans

Plans are being developed to extend this verification system to become part of an integrated Product Data Management (PDM) environment with design entry and analysis tools, configuration management tools and workflow management tools. A PDM environment allows a company to handle the rapidly growing quantity of data being generated electronically with a diversity of tools and to keep the different system elements, that are in various but different development stages, together (see figure 7).
In such a controlled configuration system the various stages of the prototype can be regenerated and compared with each other to verify the individual design steps. Different tools will be available in a corporate design environment. With the introduction of a TCP/IP interface to ARTIS prototyping elements can be physically scattered over a network and even the Internet can be used to connect elements together that are developed at different locations. Also the simulation of electronic circuit designs can run on a different platform in the network while feeding the prototype ‘test bus' with data.

Another activity being undertaken is to introduce a transition from functional model into physical implementation using VHDL. This is of particular importance for the design of Application Specific Integrated Circuits.

4 Conclusions

The paper has reported on some of the results of the work undertaken on the application of the combination of system design and test tools for the verification of avionics.

The results obtained during the introduction of the tools have led towards the establishment of an Avionics Research Test and Integration System ARTIS, which allows the incorporation of prototype components of different technologies in different stages of development.

5 References

1 Collinson, R.P.G., 1996, '*Introduction to avionics'*. Chapman & Hall, London, UK.

2. Harel, David, Hagi Lachover, Amnon Naamad, Amir Pnueli, Michl Politi, Rivi Sherman, Aharon Shtull-trauring, and Mark Trakhtenbrot, *"Statemate: A working environment for development of complex reactive systems"*, IEEE Transactions on Software Engineering, v.16, n.4, April, 1990.

3. Harel, David; Michal Politi. 1996. *Modeling reactive systems with statecharts: The StateMate approach.* i–Logix, part No. D-1100-43, 6/96

4. i-Logix, 1996. StateMate MAGNUM, User Guide, version 1.01, 1996.

5. Pecht, Micheal. 1995. *Product reliability, maintainability and supportability handbook.*CRC Prss, Inc.

6. Sahraoui, A.E.K., M. Romdhani, A. Jeffroy and P. Dechazelles. 1996. *A Multi-formalisms Approach for Codesign in the Development of Avionics*, IEEE Aerospace and Electronic Systems Magazine, May 1996, ISSN 0885-8985, Vol. 11, no. 5; 33-38.

6 Biography

Louis Aartman is project engineer and deputy head of the Electronics Department of the Electronics and Instrumentation Division of the National Aerospace Laboratory NLR in Amsterdam (The Netherlands). Since 1981 he has been involved as a system engineer in the development of electronics and avionics products, e.g. in the field of remote sensing, spaceborne experiment facilities and helicopter avionics. He is also active in the introduction of Computer Aided Engineering facilities for electronics. He is a member of the IEEE Instrumentation and Measurement Society.

Steven Wellink is software engineer at the NLR Embedded Systems Department of the Informatics Division, and was responsible for the development of several test and data acquisition systems based on ARTIS.

Rene Eveleens is system engineer at the Avionics Department of the Electronics and Instrumentation Division and among others involved in the introduction of system engineering tools (like StateMate) in the development and study programs.

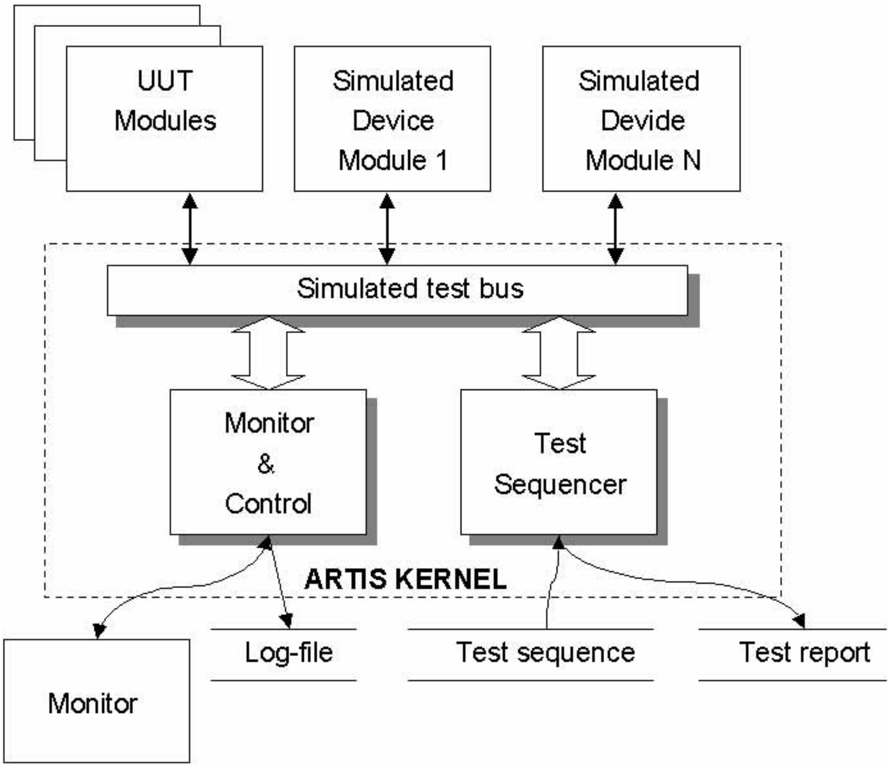

Fig. 1 ARTIS architecture

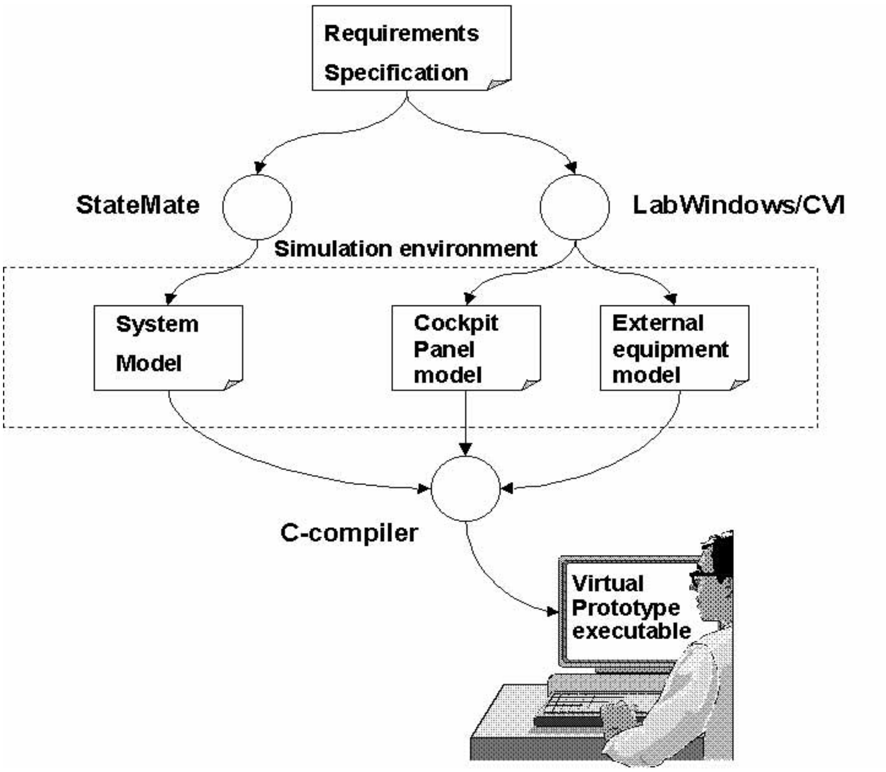

Fig. 2 From requirements to virtual prototype executable

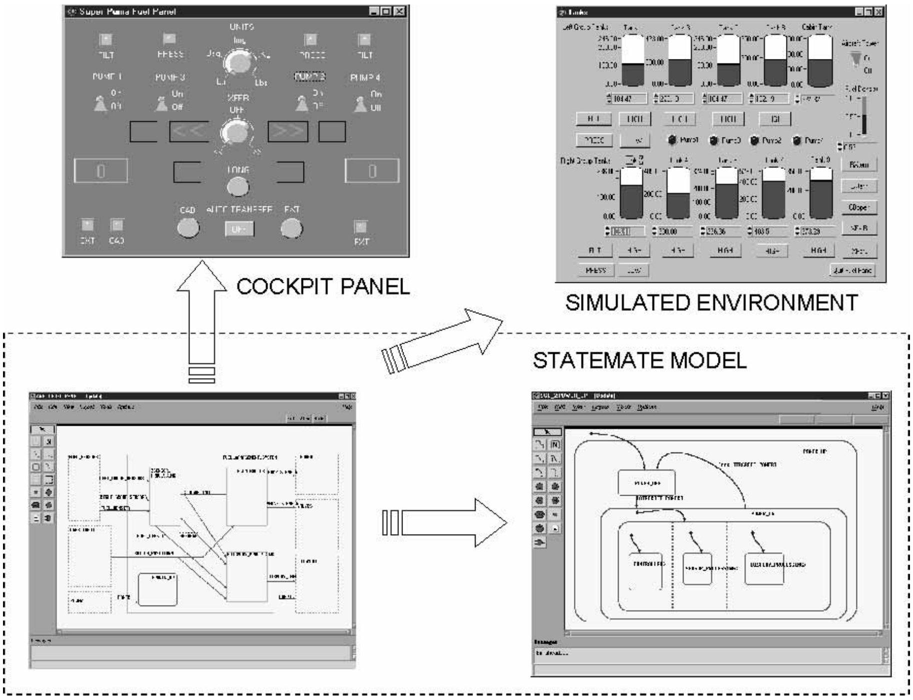

Fig. 3 Presentation of Fuel System simulation

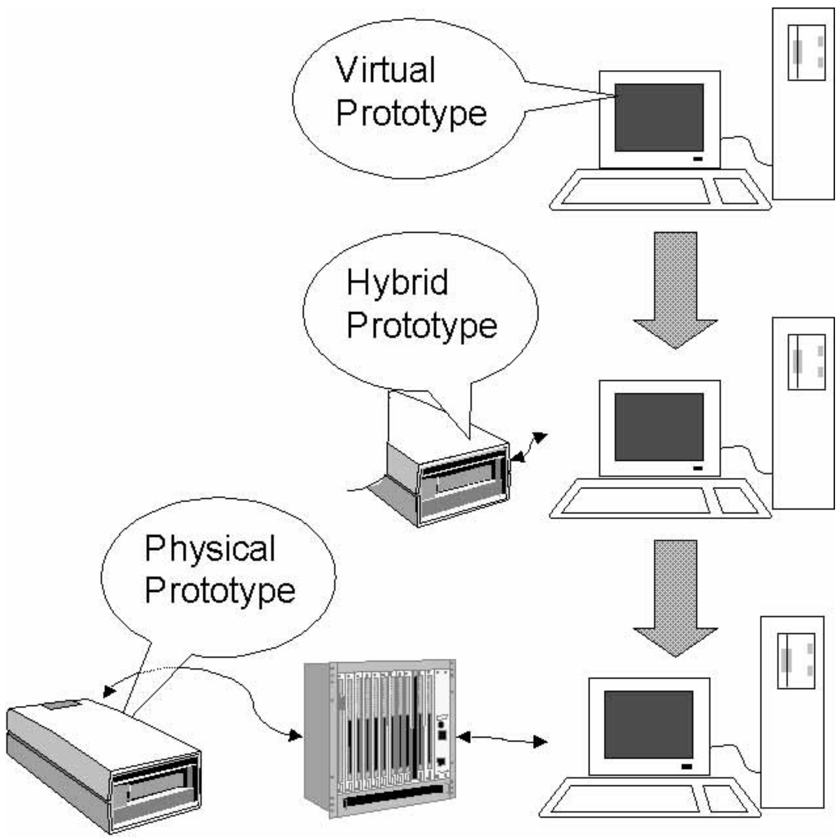

Fig. 4 Verification levels: virtual, hybrid and physical

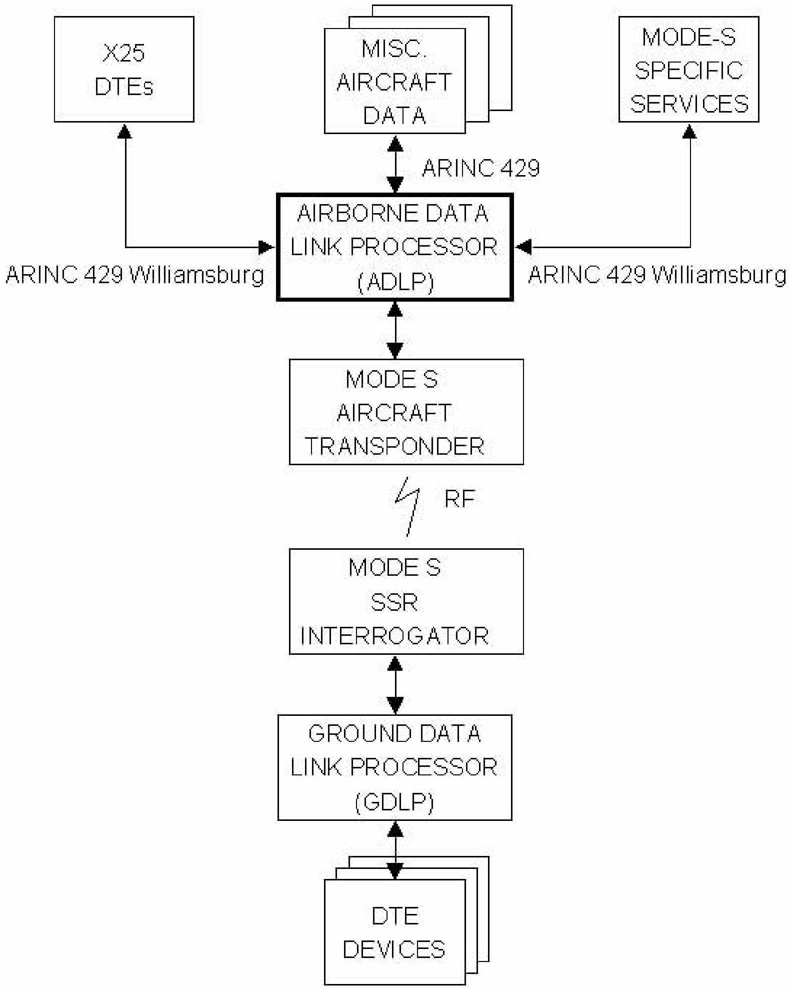

Fig. 5 The ADLP and its environment

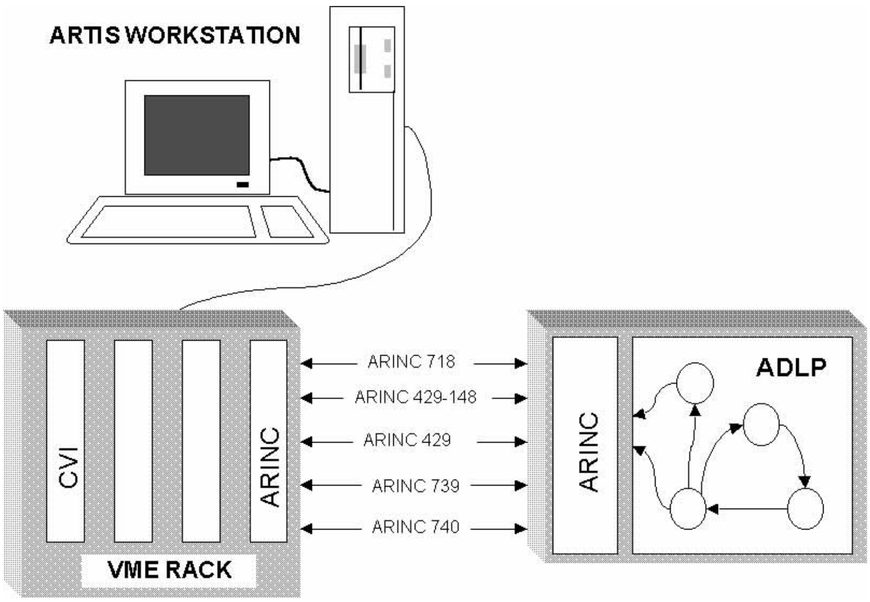

Fig. 6 Equipment involved with testing of the ADLP

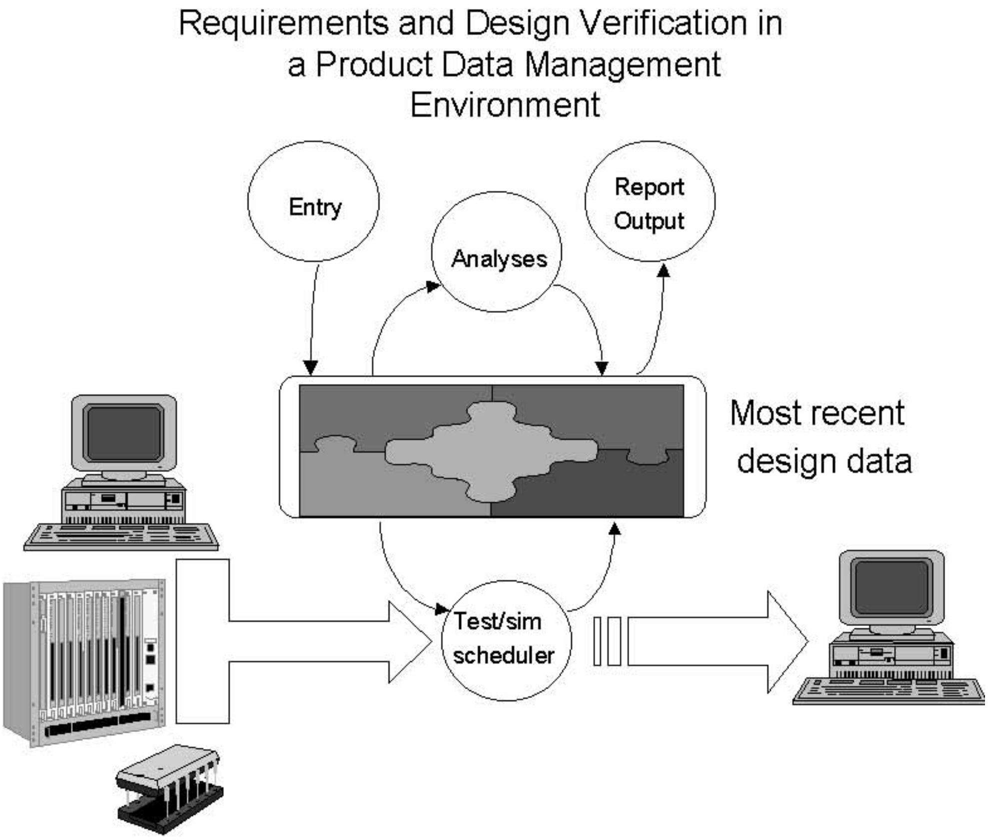

Fig. 7 Design verification in a Product Data Management environment